SPACE IMAGERY ENHANCEMENT INVESTIGATIONS; SOFTWARE FOR PROCESSING MIDDLE ATMOSPHERE DATA

Peter P. Wintersteiner
Edward Cohen

ARCON Corporation
260 Bear Hill Rd.
Waltham, MA 02451-1080

19 December 2011

Final Report

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

AIR FORCE RESEARCH LABORATORY
Space Vehicles Directorate
3550 Aberdeen Ave SE
AIR FORCE MATERIEL COMMAND
KIRTLAND AIR FORCE BASE, NM 87117-5776

DTIC COPY

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory 377 ABW Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RV-PS-TR-2011-0158 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//signed//	//signed//
Peter Crabtree, RVBYB Program Manager	Joel Mozer Chief, AFRL/RVB

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*
19-12-2011

2. REPORT TYPE
Final Report

3. DATES COVERED *(From - To)*
01 Jul 2004 to 16 Aug 2011

4. TITLE AND SUBTITLE
Space Imagery Enhancement Investigations; Software for Processing Middle Atmosphere Data

5a. CONTRACT NUMBER
FA8718-04-C-0031

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
Peter P. Wintersteiner
Edward Cohen

5d. PROJECT NUMBER
2301

5e. TASK NUMBER

5f. WORK UNIT NUMBER
849769

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
ARCON Corporation
260 Bear Hill Rd.
Waltham, MA 02451-1080

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Space Vehicles Directorate
3550 Aberdeen Ave SE
Kirtland AFB, NM 87117-5776

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RVBYB

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RV-PS-TR-2011-0158

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited. (66ABW-2010-0941, dtd 2 Aug 2010)

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report summarizes work related to optical superresolution for the ideal incoherent 1D spread function, multi-frame image registration, and high resolution image reconstruction. It also reports on certain software that has been developed to facilitate the management of middle atmosphere data from the Sounding of the Atmosphere using Broadband Emission Radiometry (SABER) instrument. The file structure required to use the codes is described, as are the codes themselves and their usage.

15. SUBJECT TERMS
SABER, TIMED, NetCDF, event data files, optical superresolution, incoherent image eigensystem, image registration, multi-frame image reconstruction, deconvolution

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Peter Crabtree
Unclassified	Unclassified	Unclassified	Unlimited	48	**19b. TELEPONE NUMBER (*Include area code)***

This page intentionally left blank.

TABLE OF CONTENTS

Approved for public release; distribution unlimited.

LIST OF FIGURES

Figure **Page**

Approved for public release; distribution unlimited.

LIST OF TABLES

Approved for public release; distribution unlimited.

INTRODUCTION

This Final Report describes work accomplished by ARCON Corporation according to the provisions of contract #FA8718-04-C-0031. Much of the completed scientific work has already been summarized in the Interim Scientific Reports that we have submitted over the years. We reference those reports below, in Section 4, which includes brief synopses of their contents.

In addition, we describe our most recent work, not the subject of any of those Interim Reports. One topic we address is image enhancement. The context of our efforts is CCD imagery at the focal plane of a telescope or camera that might be trained on distant moving objects, e.g. resident space objects (RSOs). The objective has been to improve methods of imaging and identifying such objects. This discussion comprises Section 2.

A second topic is software that we developed for the purpose of accessing, organizing, and analyzing data from the Sounding of the Atmosphere by Broadband Emission Radiometry (SABER) instrument, briefly described below. Section 3 describes codes that we have written to create so-called event data files, the basic set of ASCII files that is used when analyzing any segment of the SABER scientific database. These codes have been submitted to the Air Force as they have been completed and/or updated.

As noted above, Section 4 gives a synopsis of contract work that has already been reported.

2. IMAGE ENHANCEMENT FOR RESIDENT SPACE OBJECT CHARACTERIZATION

2.1 Background

The work performed during this period involved themes underlying space situational awareness (SSA) objectives, which require a capability for detecting and assessing potential threats posed by resident space objects (RSOs) orbiting near vital US satellites. Having high resolution (HR) images of RSO surface features offers an important way to identify and characterize such objects. In this context, one focus of our work was to better understand the potential of both superresolution (SR) imaging and multiframe HR image reconstruction (i.e., dealiasing) to provide improved imagery of RSOs. This work included the development of Matlab algorithms to process a sequence of snapshots taken with a CCD/CMOS camera in order to produce one or more HR image estimates. Section 2.2 describes our SR-related investigations, where the aim is to overcome the diffraction limit. Section 2.3 describes our multiframe HR image reconstruction studies, where the aim is to utilize subpixel displacement to achieve supersampling and thereby eliminate aliasing artifacts.

2.2 Superresolution-Related Investigations

Because of the fundamental role played by incoherent imaging with CCD systems, we investigated the properties of the ideal diffraction-limited incoherent one-dimensional point spread

Approved for public release; distribution unlimited.

function (PSF). The purpose was to see if it had a possible application to superresolution (SR) imaging. Before beginning this task, however, we reviewed the well-known situation involving the one-dimensional coherent PSF.

For one-dimensional coherent imaging with diffraction-limited optics, the prolate spheroidal wavefunctions (PSWFs) arise in the eigensystem, $\{\phi_n(x);\lambda_n\}$, associated with the *sinc* spread function response, where *sinc(z)* $\equiv \sin(\pi z)/(\pi z)$. Double orthogonality over the intervals $(-\infty,\infty)$ and $(-1,1)$ as well as other important properties of the eigensystem are well known and need not be elaborated upon here. We decided to calculate the PSWFs both to examine some of their features and to provide a check on the numerical method we eventually adopted for subsequently developed codes for both the coherent and incoherent spread functions.

The eigensystem applicable to one-dimensional diffraction-limited incoherent imagery, defined here as $\{\psi_n(x);\mu_n\}$, has not been nearly as thoroughly studied as that for coherent imagery. This system pertains to the $sinc^2(z)$ spread function. In particular we wanted to see if the eigenfunctions showed any features suggestive of SR as is the case with PSWF modes. For example, the PSWFs exhibit a spatial chirping, or frequency modulation, within the normalized spatial interval (-1,1) in addition to double orthogonality. If the $sinc^2$ *(z)* eigenmodes showed similar behavior, this would suggest a capability for superresolution with potential practical utility. Our efforts turned to the calculation of both eigensystems.

2.2.1 Eigensystem Formulations

An analysis and method for calculating the eigensystems of general bandlimited functions, such as the *sinc* and $sinc^2$ kernels, was described by Khare and George [1] using the sampling theorem. We adopted their method initially but went on to develop an algorithm based on a more direct numerical method. The latter was initiated with the intention of performing faster computation while maintaining high accuracy.

The eigensystems are defined by the solutions to the following Fredholm integral equations of the 2nd kind:

$$\lambda_m\phi_m(u) = 2BL\int_{-1}^{1} dv\cdot \mathrm{sinc}[2BL(u-v)]\phi_m(v) \qquad [1]$$

for coherent imagery and

$$\mu_n\psi_n(u) = 2BL\int_{-1}^{1} dv\cdot \mathrm{sinc}^2[2BL(u-v)]\psi_n(v) \qquad [2]$$

for incoherent imagery. In dimensionless variables, $u = x/L$ and $v = x'/L$. Here, 2L is the aperture width and (-B,B) is the spatial frequency range of the bandlimited image being sampled.

In this notation the double orthogonality satisfied by the PSWFs is given by

$$\int_{-1}^{1} dv \cdot \phi_m(v)\phi_n(v) = \lambda_n \delta_{mn} \qquad [3]$$

and

$$\int_{-\infty}^{\infty} dv \cdot \phi_m(v)\phi_n(v) = \delta_{mn}, \qquad [4]$$

with δ_{mn} being the Kronecker delta function. The $sinc^2$ eigenfunctions, $\{\psi_n\}$, were verified not to satisfy double orthogonality.

2.2.2 Numerical Method of Solution

To calculate the eigensystems $\{\psi_n(x);\mu_n\}$ and $\{\phi_n(x);\lambda_n\}$ using a direct numerical approach, we used the Matlab *eig* function. First, this required a system matrix be formed in each case by discretizing each linear integral operator equation; one based on the *sinc* and the other the $sinc^2$ kernels. For discretization a collocation scheme was developed that involved dividing the object integration range (-1,1) into M equally spaced intervals. It was assumed that the eigenfunction within a given interval was approximately uniform and could be factored out of each sub integral. The kernel, e.g., $sinc^2$, was then integrated over each interval using a low-order Gauss quadrature, typically 4-point. M = 201 was used for most of our calculations although larger and smalller values of M were used to test the accuracy of the computations. The matrix formed in both cases had a real symmetric Toeplitz form, $S_{|k-j|}$. Because of this structure, each full matrix required explicit calculation of only the 1st row of matrix elements. A full Toeplitz matrix was then conveniently generated using Matlab's *toeplitz* function. Finally, the matrix-vector eigensystem was computed using the *eig* function.

2.2.3 Numerical Results and Conclusions

Figure 1 shows the eigenvalue distribution μ_n vs n associated with the $sinc^2$ eigenfunctions for the diffraction-limited 1-D incoherent PSF. The Shannon number $c = \pi 2BL = 5$ was arbitrarily chosen for this case and the distribution suggests a triangular form. Figures 2-5 show representative modes for which mode 7 or 8, and higher, are associated with very small eigenvalues.

With $c = 20$ a stronger triangular shaped eigenvalue curve is evident in figure 6. In all cases, one finds a smooth decay as the eigenvalues approach zero. While μ_n vs n qualitatively resembles the slit aperture optical transfer function, the latter is a true triangle function with a sharp cutoff. Figures 7-10 show representative eigenmodes for $c = 20$. Figures 11 and 12 show the approximate expansion of a rectangular pulse function using the $sinc^2$ eigenfunctions for different values of *c*. Severe overshooting results at the edge discontinuities as expected, but in both cases the images remain non-negative.

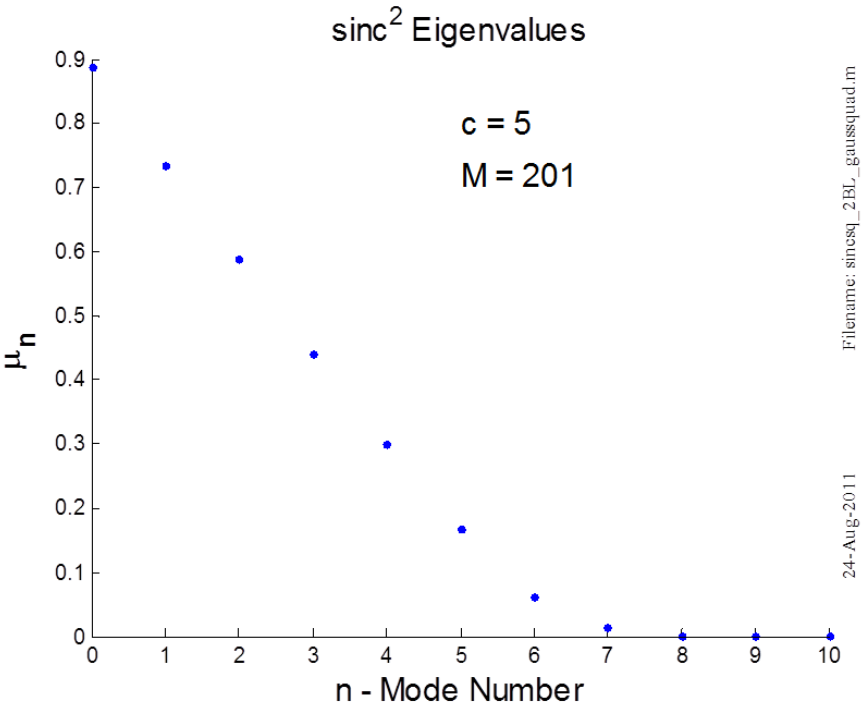


Figure 1. Incoherent Spread Function Eigenvalues (c = 5)

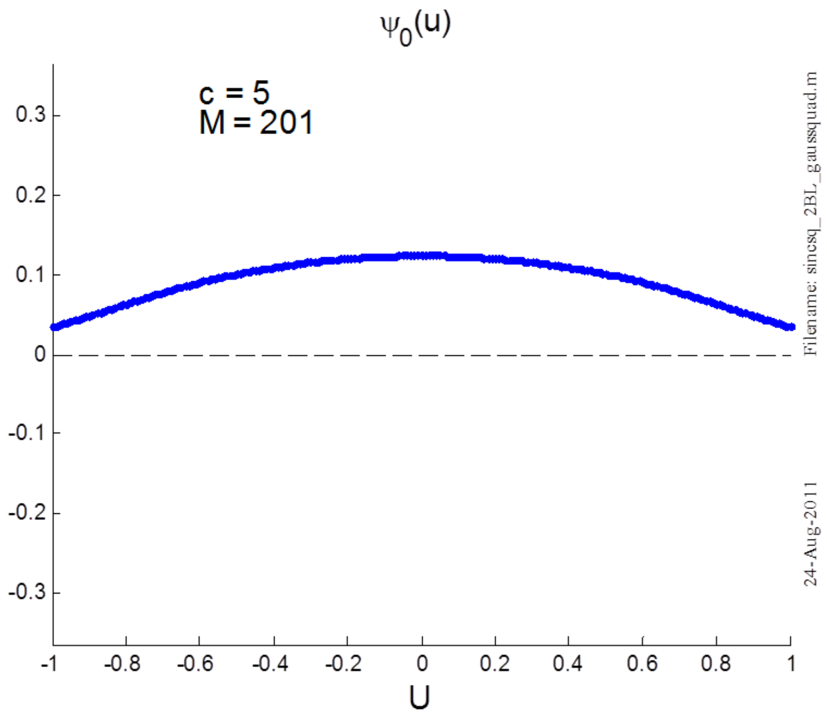


Figure 2.   Incoherent Eigenmode 0 (c = 5)

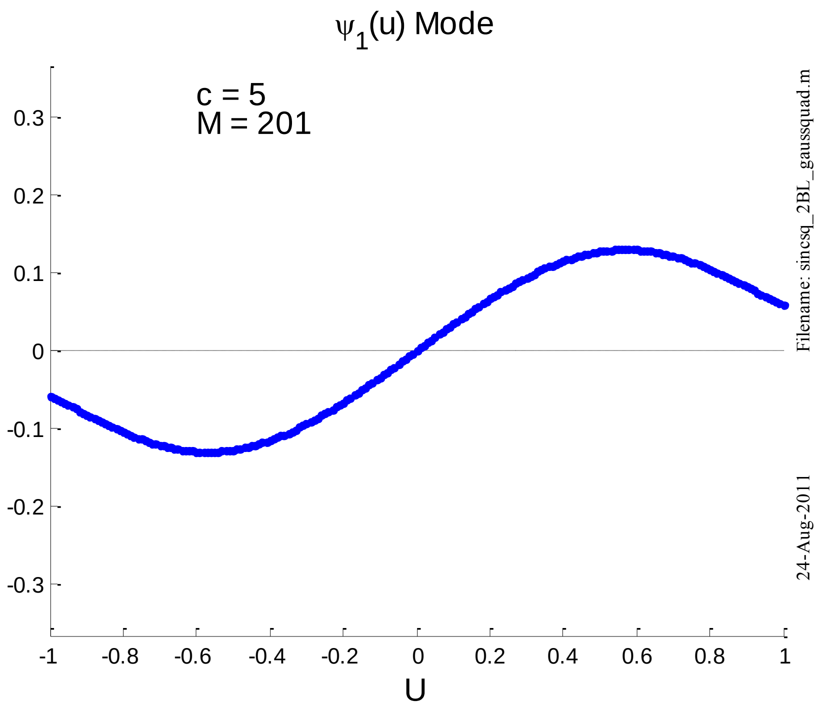


Figure 3. Incoherent Eigenmode 1 (c=5)

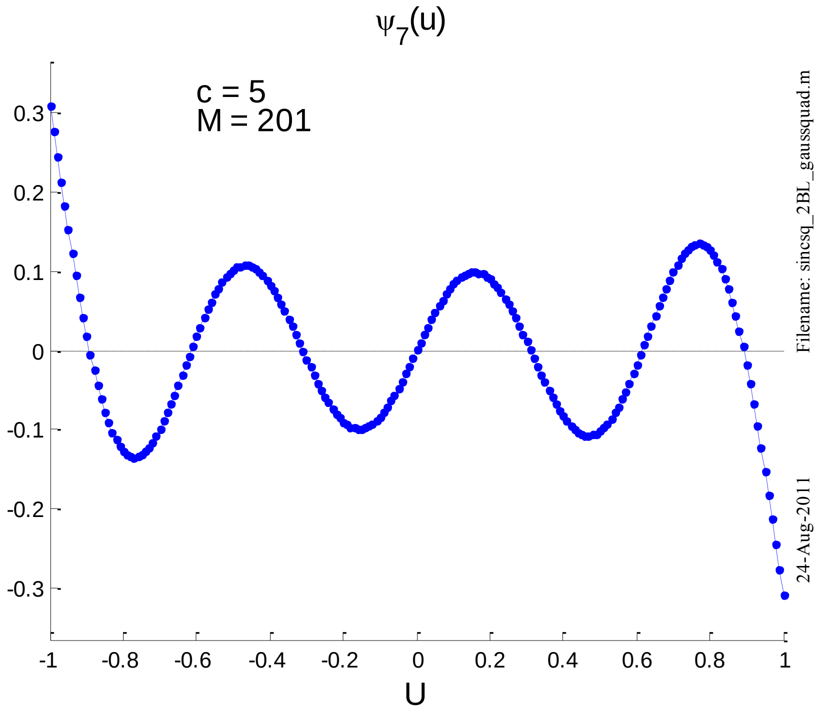


Figure 4. Incoherent Eigenmode 7 (c=5)

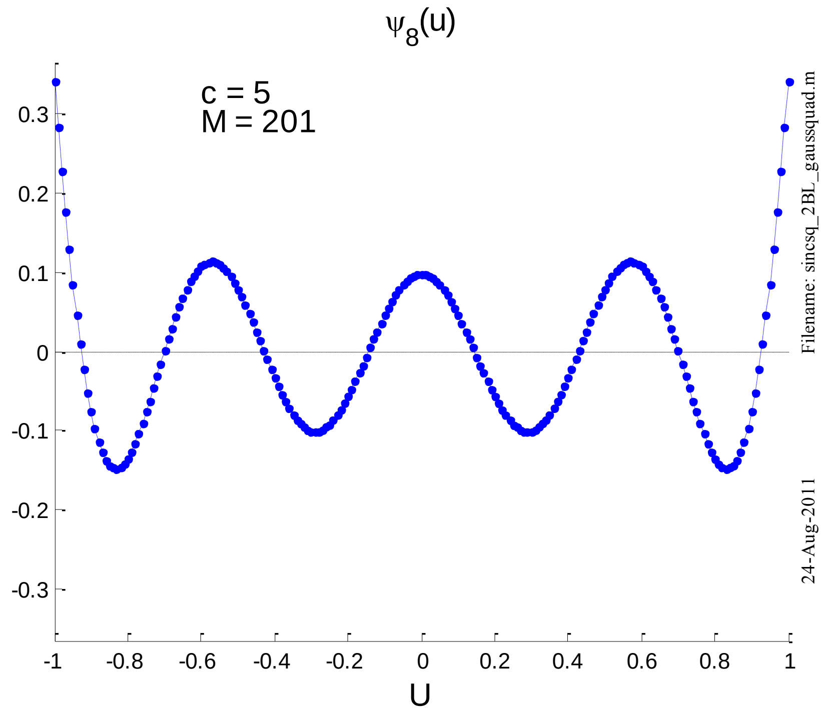


Figure 5. Incoherent Eigenmode 8 (c=5)

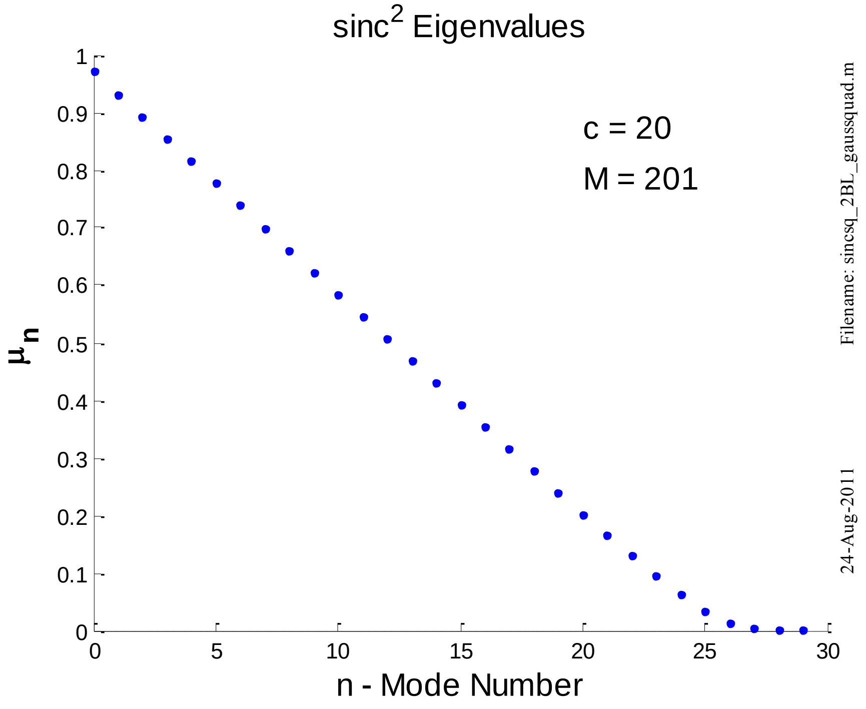


Figure 6. Incoherent Spread Function Eigenvalues (c = 20)

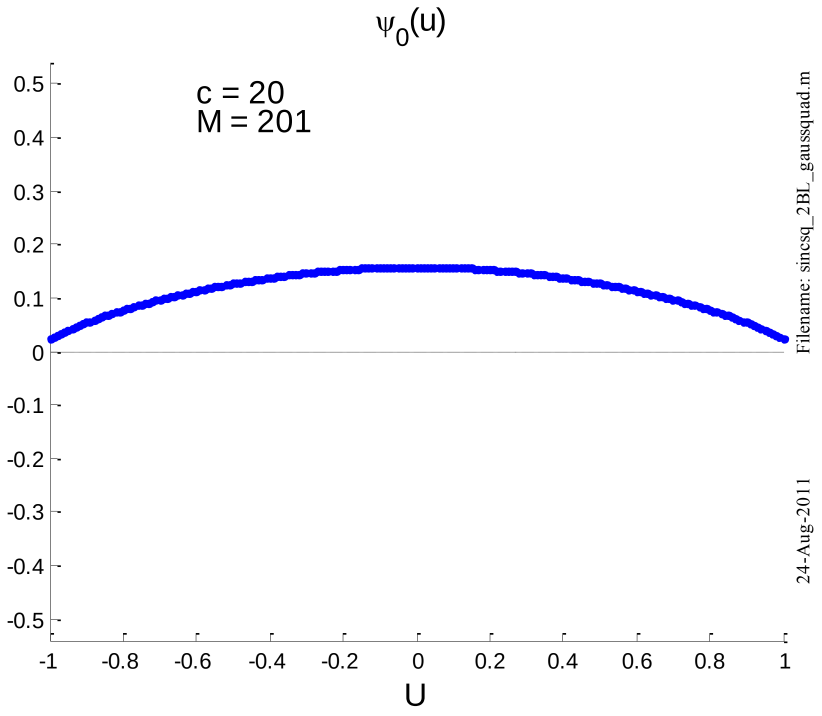


Figure 7. Incoherent Eigenmode 0 (c = 20)

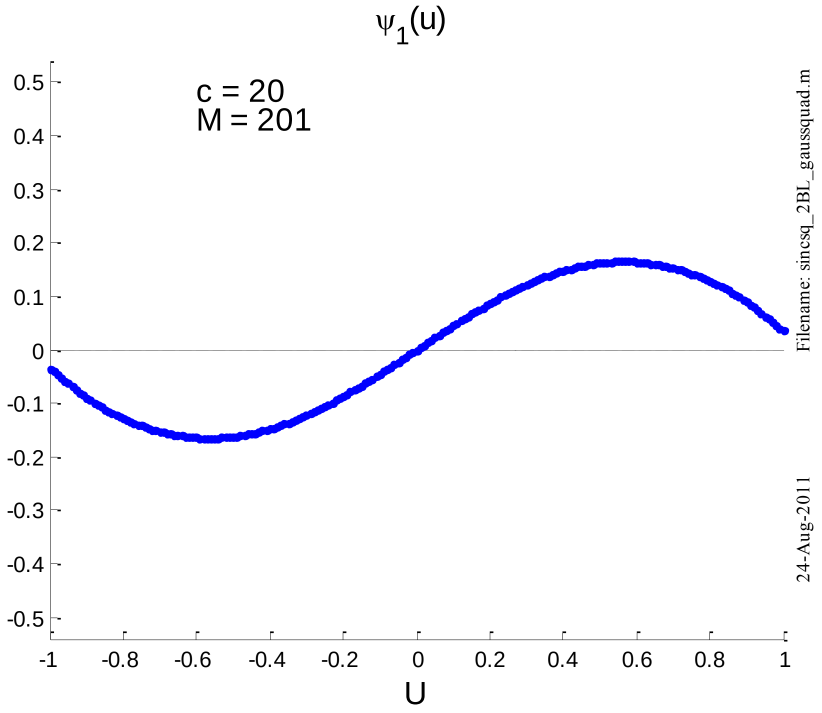


Figure 8. Incoherent Eigenmode 1 (c = 20)

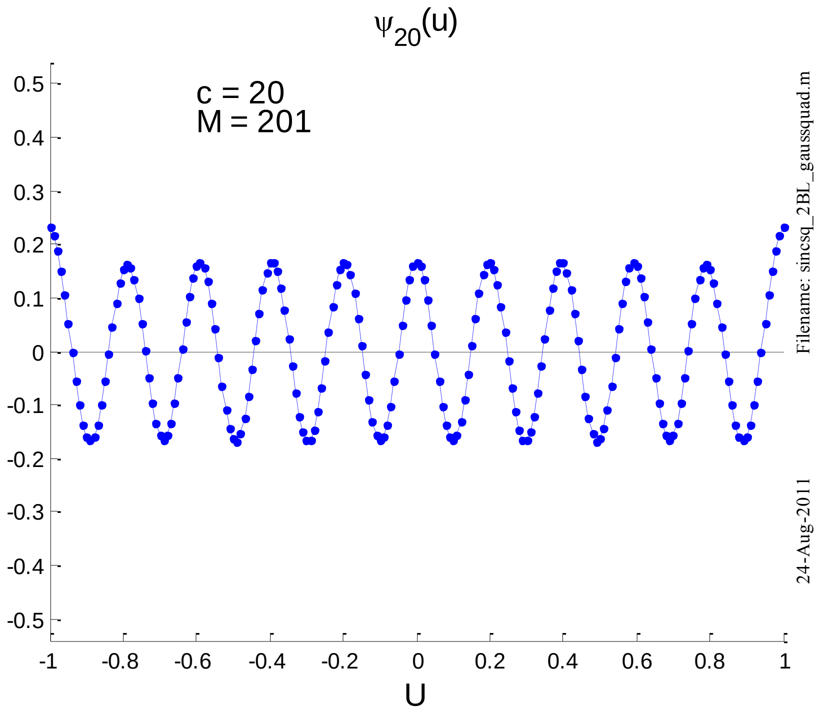


Figure 9. Incoherent Eigenmode 20 (c = 20)

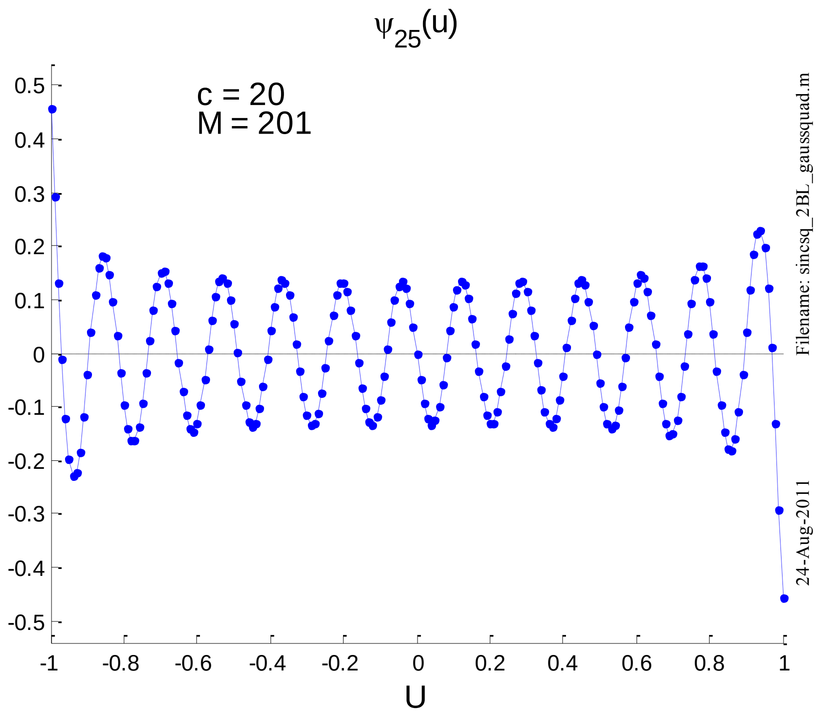


Figure 10. Incoherent Eigenmode 25 (c = 20)

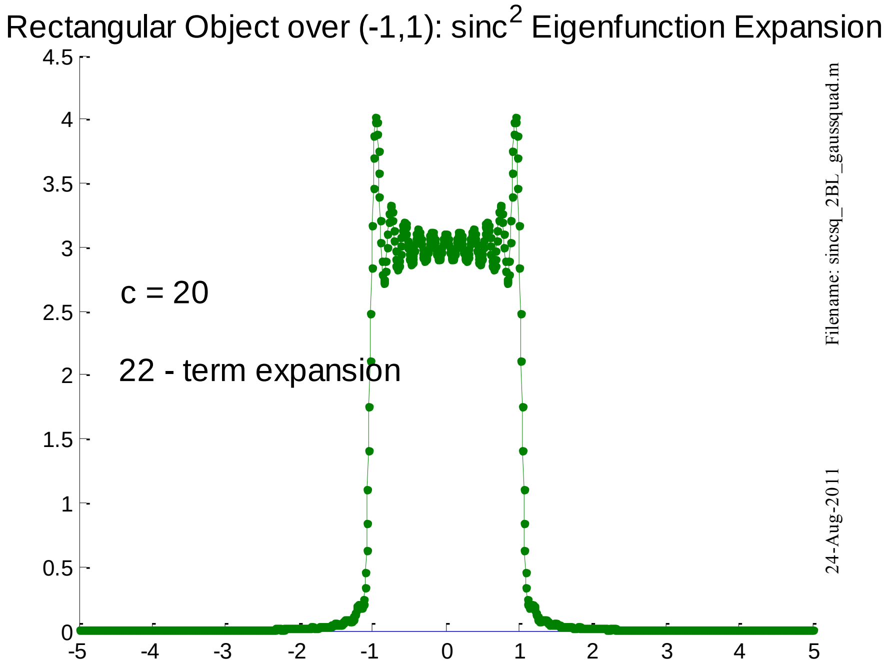


Figure 11. Moderate Bandwidth Rectangular Object Expansion

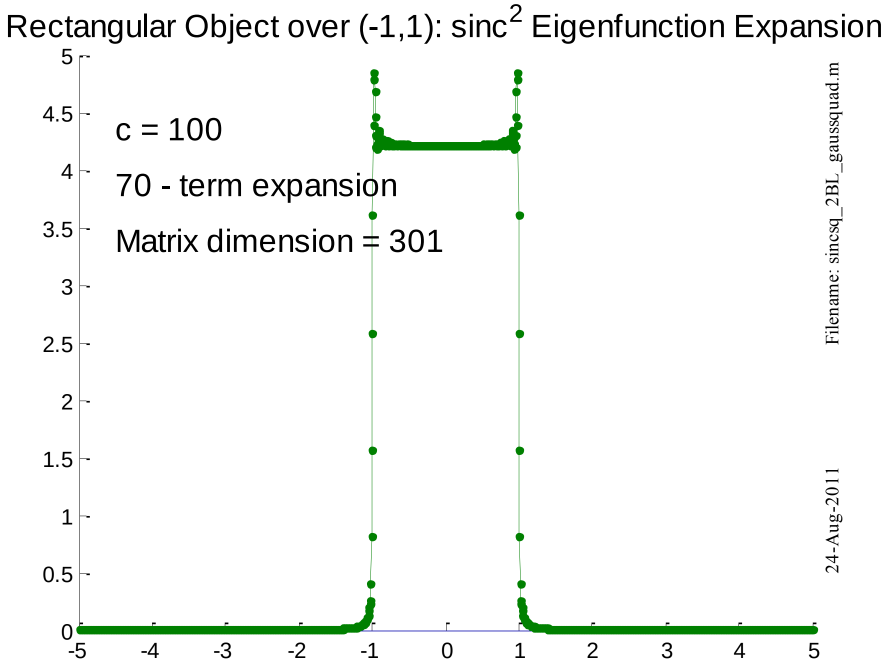


Figure 12. Large Bandwidth Rectangular Object Expansion

We then tested a given mode for spatial chirping by calculating zero-crossings within (-1, 1). The average value of the zero-crossing separations provided a measure of the half-period, $T/2$, for that mode. The frequency difference

$$\Delta f = {}^{1}\!/_{T} - 2B \qquad [5]$$

was calculated and positive values were understood as being indicative of SR. We found that this occurred only for sufficiently small eigenvalues. For example, using $c = 5$ our computations showed that with a PSWF expansion, $\Delta f > 0$ appeared for modes ≥ 3. $\Delta f > 0$ occurred for $sinc^2$ modes ≥ 7 in the incoherent imaging case. Numerically, $\lambda_3 = 0.3509$ for PSWF mode 3, whereas $\mu_7 = 0.0142$ for the *sinc*^{2} mode 7. In general, determining unfiltered $g(x)$ expansion coefficients by invoking orthogonality leads to the coefficients being inversely proportional to the eigenvalues. One measure of the relative noise sensitivity in this example is the ratio $\lambda_3 / \mu_7 \approx 25$ where the 1st super-resolving modes occur. This suggests that sensitivity to additive spatial noise for all super-resolving modes is considerably greater for the *sinc*^{2} expansion coefficients than PSWF expansion coefficients.

For a larger value, $c = 20$, SR begins to appear at mode 26 where $\mu_{26} = +0.01279$ and $\Delta f = 0.0125$ for $sinc^2$ modes. Using a PSWF expansion instead, SR modes develop at $\lambda_{13} = 0.2473$ where $\Delta f = 0.1009$. A measure of the relative noise sensitivity is now given by $\lambda_{13} / \mu_{26} \approx 19$. For higher-order mode numbers, the eigenvalues again diminish rapidly in both situations while Δf correspondingly increases. We conclude that under ideal 1-D conditions with incoherently illuminated objects, SR image content in the presence of noise will be more poorly represented with expansions based on $sinc^2$ eigenmodes than with PSWFs.

2.3 Multiframe High Resolution Image Reconstruction

2.3.1 Image Registration

The first stage of this effort to untangle aliased spatial frequencies in CCD images required developing an algorithm for registering multiple low resolution (LR) snapshots of moving space objects. Registered images were then to be dealiased and finally deconvolved for HR image reconstruction. Both a linear and a nonlinear method of reconstruction were pursued for these objectives. SR as used herein means recovering aliased spatial frequencies in LR images caused by pixel-limited sampling of a focal plane array. In particular this investigation assumed the Nyquist spatial frequency of the array was much lower than the cutoff frequency of the camera lens optical transfer function.

After surveying various publications on registration algorithms, the spatial domain analysis and pseudo-code described by Hardie et al [2] for image registration and high resolution (HR)

Approved for public release; distribution unlimited.

image reconstruction was selected as a starting point for our investigations and algorithm development. An attractive feature of the formulation described in [2] was that it could be applied to objects exhibiting (h)orizontal and (v)ertical relative displacements, as well as rotations (θ) with respect to the camera. A key assumption was that (h,v,θ) were caused by the relative motion of an object translating and rotating in a plane perpendicular to the optical axis of the camera system. The approach appeared to be thorough both with regard to physical formulation and mathematical analysis; however, as we were to discover, several key omissions or ambiguities caused problems that needed to be addressed in order to actually perform meaningful registration.

The registration method [2] considered a sequence of *p* images of an RSO moving with respect to a camera. The goal of image registration was to bring all *p* image intensity distributions into geometric coincidence. Once all *p* images were registered, an approximate HR reconstruction of the original object would be made.

Registration itself required motion predictions of (*p*-1) images relative to, say, the 1st snapshot which served as a fixed reference image. It was assumed in our work that the snapshots were imaged onto an ideal CCD detector array. For specificity we focused on the particularly important problem of frame-to-frame sub-pixel motion estimation. Issues such as image acquisition, spatially variant point spread functions, motion blur, etc. were not considered.

2.3.2 Motion Prediction for Image Registration

The registration process detailed in reference [2] involved minimizing the squared error between the k^{th} image, defined in its coordinate system (*x*,*y*), and a reference image defined in its own coordinate system, (x'', y''). This required solving an unconstrained optimization problem for the (h,v,θ) by minimizing the cost function, C_k:

$$C_k = \sum_{\mathrm{R}} \left\{ I_1(x", y") - I_k(x, y) \right\}^2 \quad \mathrm{k = 2,3,...,p} \qquad [6]$$

where

$I_1(x", y") =$ intensity of LR reference image observed in its coordinate system;

$I_k(x, y) =$ intensity of k^{th} LR image observed in the (moved) camera coordinate system;

R = region where reference image pixels and k^{th} image pixels overlap.

Coordinate relationships between the two systems are given by

$$x'' = x\cos(\theta) - y\sin(\theta) + h; \qquad [7]$$

$$y'' = x\sin(\theta) + y\cos(\theta) + v. \qquad [8]$$

In general this leads to a nonlinear least-squares formulation for determining the (h,v,θ) defined by C_k. The formulation was linearized as in [2] by considering small image rotation angles, θ, using a 3-term Taylor series approximation for $I_1(x", y")$. C_k was then minimized with respect to (h,v,θ) and the normal equations solved.

Although the 3-term expansion permits one to solve a linear least-squares problem, this low level of approximation is very poor except near the expansion point chosen to be at the image center. Using more terms in the expansion would under ideal conditions improve the image representation; however, numerical approximations of higher-order derivatives generate their own significant errors and amplify image noise. A higher-order expansion in (h, v, θ) would also lead to another nonlinear optimization problem. After much experimentation we stayed with the 3-term expansion and settled mostly on Prewitt operators for a smoothed gradient approximation.

A significant problem arose that was not discussed in [2] that had a major deleterious effect on our registration algorithm accuracy. Upon close numerical examination of the matrix elements defining the normal equations shown in equation (30) of reference [2], we found that the matrix was dominated by contributions from pixels increasingly distant from the point about which the Taylor series was expanded. In this regard, the term from equation (27) of [2] involved the formula

$$\bar{g}(\mathbf{n}) = n_1 T_1 \hat{g}_y(\mathbf{n}) - n_2 T_2 \hat{g}_x(\mathbf{n}), \qquad [9]$$

which varies linearly with $x = n_1 T_1$ and $y = n_2 T_2$. Here, $\mathbf{n} = (n_1, n_2)$ are pixel indices and T_1, T_2 are pixel separations along the respective coordinate axes. For LR images having tens to thousands of pixels in each dimension, $\bar{g}(\mathbf{n})$ dominates other matrix elements as $|n_1|$ or $|n_2|$ increases unless the image intensities are strongly attenuated at those particular large n_1, n_2 pixel values. This suggested to us that some kind of weighting or tapering down of the intensities away from the origin would improve registration efforts. To this end we sought to improve our registration algorithm using a Gaussian weighted least-squares formulation. The cost function was thus modified to

$$C_k = \sum_{\mathrm{R}} E_k(x,y) \cdot \left\{ I_1(x", y") - I_k(x,y) \right\}^2 \qquad [10]$$

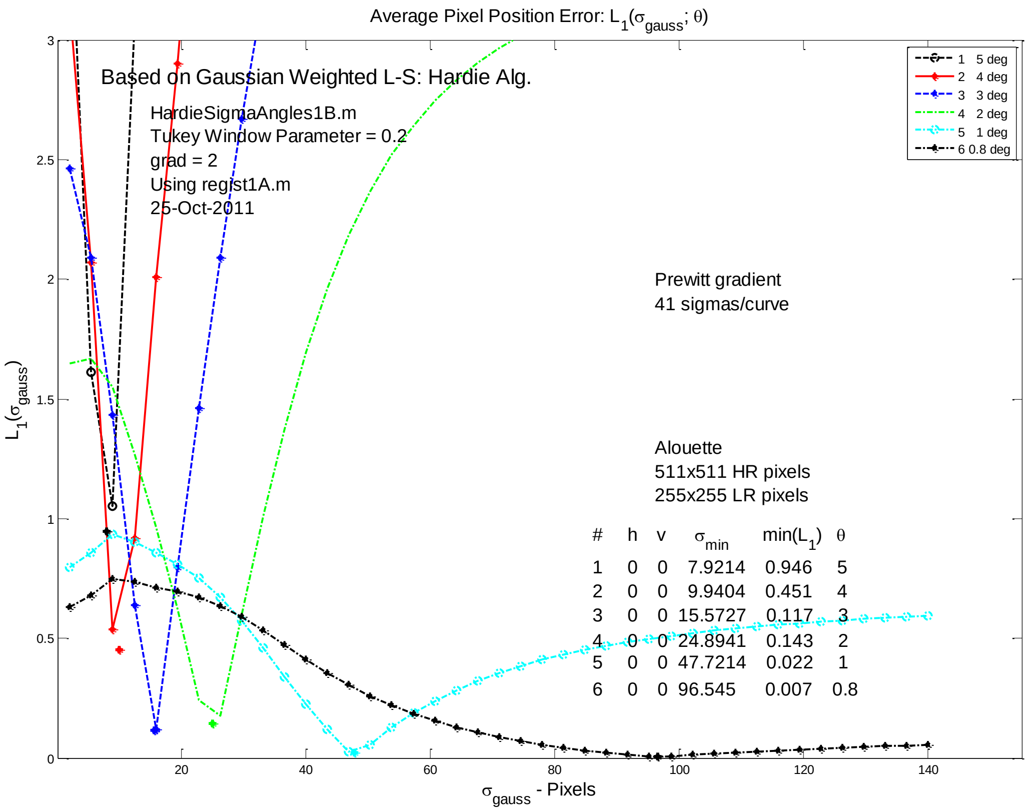
where

$$E_k(x,y) = \exp(-\frac{x^2 + y^2}{2\sigma_k^2}). \qquad [11]$$

To quantify the results of the registration, an error metric was required. L_1 and L_2 were both calculated in the codes. L_1 always gave a larger value which we viewed as being a more conservative error estimate. Because it also appeared in various papers we read at the time, it was adopted as the metric for defining the average pixel position error of each registered image, viz.,

$$L_1 = \frac{1}{\#\ of\ \ nonNaNs}\left(\sum_{\substack{m,n\\ nonNaNs}} \left[|\,\Delta x_{m,n}\,| + |\,\Delta y_{m,n}\,| \right] \right), \qquad [12]$$

where *nonNaNs* indicate overlapping pixels. Other than that, there appears be no compelling reason to use one over the other.



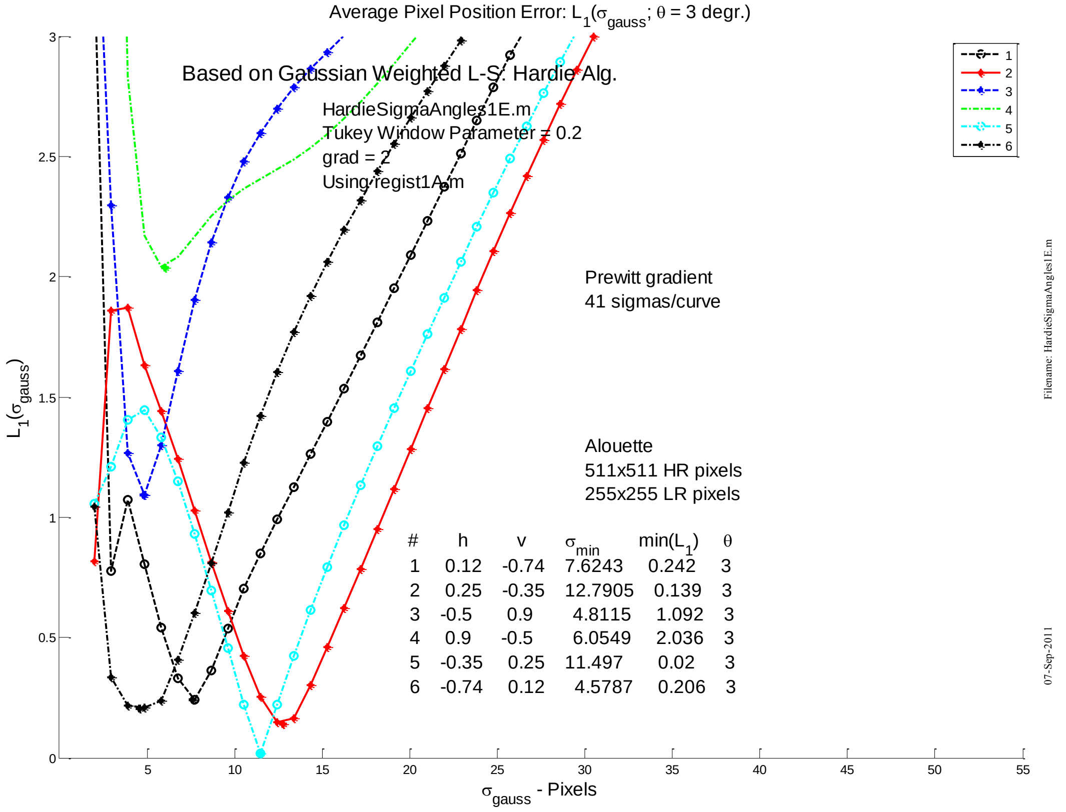
#	h	v	σ_{min}	min(L$_1$)	θ
1	0	0	7.9214	0.946	5
2	0	0	9.9404	0.451	4
3	0	0	15.5727	0.117	3
4	0	0	24.8941	0.143	2
5	0	0	47.7214	0.022	1
6	0	0	96.545	0.007	0.8

Figure 13. Pixel Position Errors for Image Rotations

As seen from figures 13 and 14, which are examples representative of many image motions we considered, there is an optimal value of σ_k that minimizes L_1. The problem is that the optimal choice for σ is not known *a priori*. It depends on (h, v, θ) for each moved image and appears to require a search for σ_k such that

$$\sigma_k^{opt} = \underset{\{\sigma_k;\, h, v, \theta\}}{\arg\min}(L_1). \qquad [13]$$

The procedure we adopted was to calculate L_1 using a set of σ-values from which parabolic interpolation was applied to find $L_1^{\min}$. In some situations, multiple minima occurred. We concluded after analysis and experimentation that the largest of the σ_k^{opt} corresponded to the physically most meaningful value; namely, the one for which the Gaussian weighting is weakest or closest to being uniform across an image. Apart from registration accuracy considerations, an apparently unavoidable drawback of this approach is that it reduces the speed of the algorithm making the overall registration method inherently inefficient.



#	h	v	σ_{min}	min(L$_1$)	θ
1	0.12	-0.74	7.6243	0.242	3
2	0.25	-0.35	12.7905	0.139	3
3	-0.5	0.9	4.8115	1.092	3
4	0.9	-0.5	6.0549	2.036	3
5	-0.35	0.25	11.497	0.02	3
6	-0.74	0.12	4.5787	0.206	3

Figure 14. Pixel Position Errors for Image Rotations and Shifts

Despite what appeared initially to be a thorough and complete analysis presented in reference [2], the noted omissions created serious difficulties and led to mediocre results. These were uncovered when our answers 1) departed significantly from "truth motions" (h,v,θ)'s with synthetic imagery; 2) did not show convergence behavior expected of the algorithm; and 3) while giving fairly good estimates for (h,v) translations only, usually gave very poor results for the general case involving (h,v,θ).

Because of such problems, a search was made for a more robust and accurate registration algorithm. This eventually led us to an existing superresolution program [3]. The *keren* function in that program was appealing because of the Gaussian pyramid structure around which registra-

Approved for public release; distribution unlimited.

tion at progressively finer resolution scales were obtained. It avoided the problem of multiple minima encountered with the unconstrained minimization algorithm of [2]. Modifications had to

Table 1. Alouette Truth Image Motions

h(true)	v(true)	deg(true)
0.629	-0.716	2.060
0.812	-0.156	-4.682
-0.746	0.831	-2.231
0.827	0.584	-4.538
0.265	0.919	-4.029
-0.805	0.311	3.235
-0.443	-0.929	1.948
0.094	0.698	-1.829
0.915	0.868	4.502
0.930	0.357	-4.656
-0.685	0.515	-0.613
0.941	0.486	-1.184
0.914	-0.216	2.655
-0.029	0.311	2.952
0.601	-0.658	-3.131

Table 2. Tolerance A Registration Errors, A = 10^{-3}

delta h	delta v	delta deg.	L1 error
-0.014	-0.010	0.029	0.066
0.067	0.011	-0.061	0.149
0.012	0.015	-0.032	0.073
0.067	0.005	-0.065	0.156
0.037	0.013	-0.069	0.155
-0.051	-0.007	0.050	0.121
-0.024	-0.005	0.035	0.080
0.016	0.012	-0.048	0.107
-0.046	0.001	0.006	0.052
0.059	-0.007	-0.071	0.164
-0.008	0.007	-0.064	0.141
0.021	0.001	-0.041	0.093
-0.021	0.009	0.040	0.090
-0.031	-0.008	0.007	0.041
0.045	-0.008	-0.053	0.124

Table 3. Tolerance B Registration Errors, B = 10^{-4}

delta h	delta v	delta deg.	L1 error
-0.014	-0.012	-0.027	0.063
0.068	0.014	0.011	0.084
0.010	0.015	0.025	0.057
0.066	0.008	-0.004	0.074
0.036	0.014	-0.003	0.050
-0.051	-0.007	-0.022	0.075
-0.024	-0.006	-0.029	0.068
0.015	0.014	0.027	0.062
-0.045	0.001	-0.005	0.051
0.059	-0.004	-0.000	0.063
-0.008	0.008	0.029	0.064
0.021	0.004	0.036	0.081
-0.021	0.006	-0.019	0.048
-0.031	-0.009	-0.022	0.059
0.046	-0.005	0.025	0.074

Table 4. Tolerance C Registration Errors, C = 10^{-5}

delta h	delta v	delta deg.	L1 error
-0.014	-0.012	-0.007	0.026
0.068	0.014	0.002	0.082
0.010	0.015	-0.001	0.025
0.066	0.008	0.001	0.074
0.036	0.014	0.000	0.050
-0.051	-0.007	-0.009	0.062
-0.024	-0.006	-0.007	0.032
0.014	0.014	0.002	0.028
-0.046	0.001	0.002	0.049
0.059	-0.004	0.004	0.064
-0.008	0.008	0.016	0.036
0.020	0.004	0.013	0.036
-0.021	0.007	-0.006	0.030
-0.031	-0.009	-0.009	0.043
0.045	-0.006	0.003	0.051

Approved for public release; distribution unlimited.

be made to the particular *keren* function we obtained because it would crash when attempting to register rotated images. It also had superfluous calls to FFTs. Results for typical runs using our rotation-corrected and more efficient version of the function are shown in Tables 1-4. Table 1 shows 15 cases involving random sub-pixel shifts and rotations within (-5,5) degrees. Tables 2-4 display deviations from truth image shifts, rotations, and average pixel position errors for convergence tolerances of $10^{-3}, 10^{-4},$ and 10^{-5} respectively. While the accuracy was found to improve with tighter tolerances, the program execution slows down as tolerances are tightened. Nevertheless, this is the most accurate and robust algorithm we currently have at our disposal. At this time we do not consider improvements to the function completed; however, the data obtained thus far is very encouraging. Time limitations prevented our pursuing this further or from developing Matlab algorithms for other registration methods.

2.3.3 Linear High Resolution Reconstruction

Having a capability for registering each snapshot of a multi-frame sequence by using, say, the modified *keren* algorithm, the next step was to dealias the images onto a single HR grid and then deconvolve the result. We decided to pursue this in several stages depending on the complexity of image motions. A key step in this procedure was determining the intensity contribution from each LR pixel that overlapped a single HR grid of pixels, using all registered LR images. For factor-of-two dealiasing and arbitrary image displacements, a minimum of 4 and maximum of 9 HR pixels can be overlapped by each LR pixel.

To test the dealiasing process, an exact and ideal situation was considered first using 4 half pixel displaced images. According to analysis provided by Zalevsky and Mendlovic [4], perfect dealiasing should result when 4 LR image shifts of the amounts [0,0], [0,0.5], [0.5,0], and [0.5,0.5] are used. The dealiased image shown in figure 15 was obtained for the test case of a simulated binary star image with a separation of exactly 2 (high resolution) pixels. As noted in section 2.3.4, the Wiener and Lucy-Richardson (L-R) deconvolved images shown in figures 16 and 17 agree very well with the binary star truth image, as did other objects considered such as those shown in figures 18-23. However, figure 24 shows the complete failure of Wiener deconvolution on a dealiased image with more spatial features than the binary star image, but for the case of non-ideal, random shifts. The failure appears to be caused by the use of random, non-ideal subpixel shifts. A perfect pixel response function, pprf, was assumed for all tests, and for factor-of-two aliasing the pprf takes the form

$$\text{pprf} = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}, \qquad [14]$$

which should be divided by 16 for unit area normalization.

We also investigated the case of uniformly distributed random sub-pixel shifted images. Exact algebraic formulas for the overlap areas needed for dealiasing factor-of-two aliased images

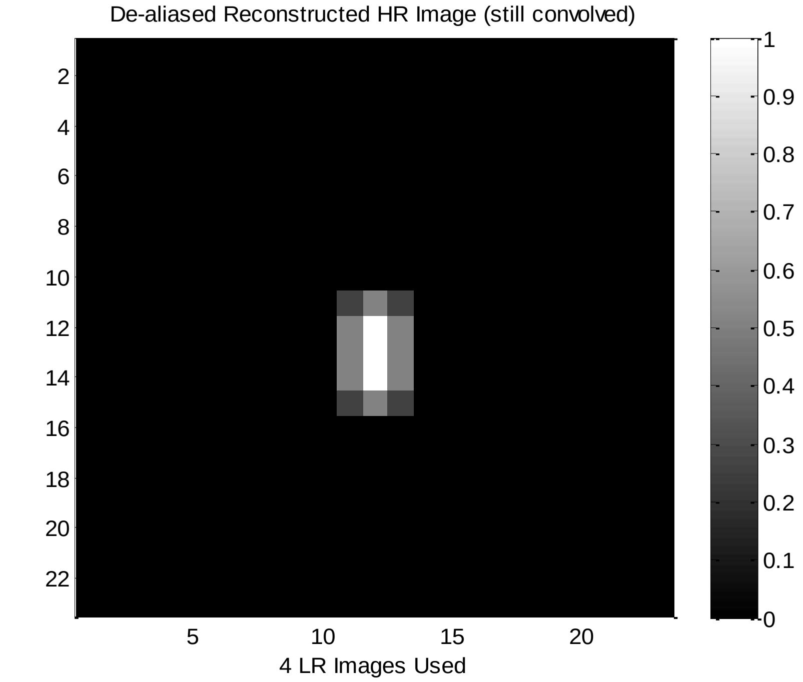


Figure 15. Binary Star Ideally and Exactly Dealiased

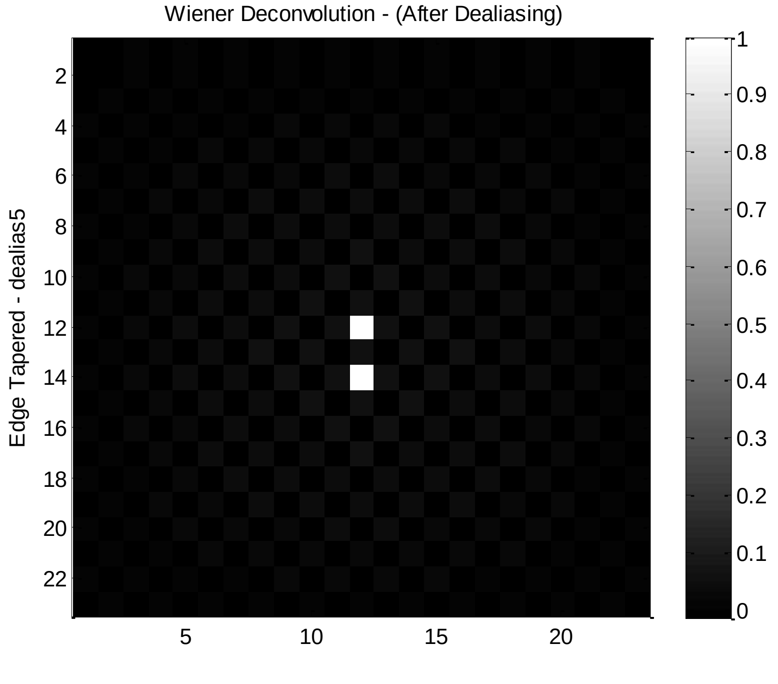


Figure 16. Wiener Deconvolution of Binary Star

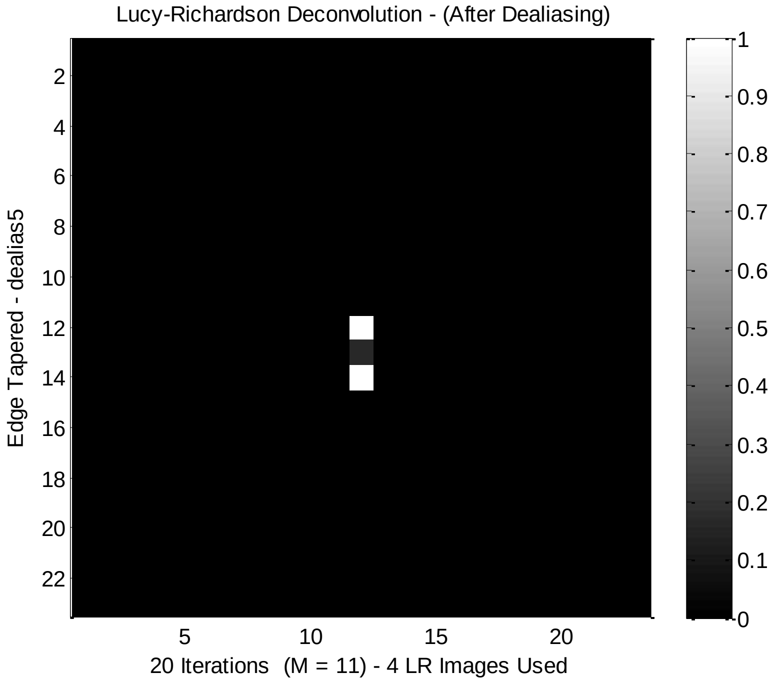


Figure 17. Lucy-Richardson Deconvolution of Binary Star

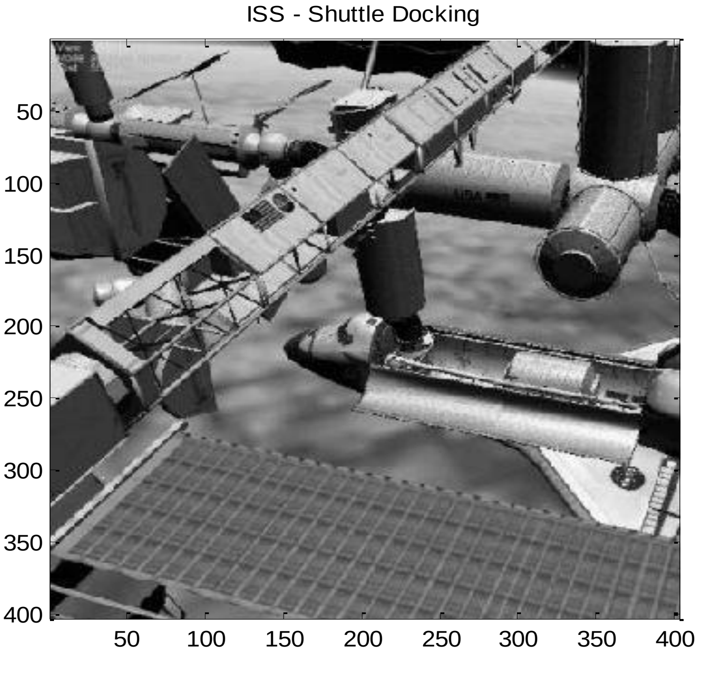


Figure 18. High Resolution ISS - Shuttle Docking Truth Image

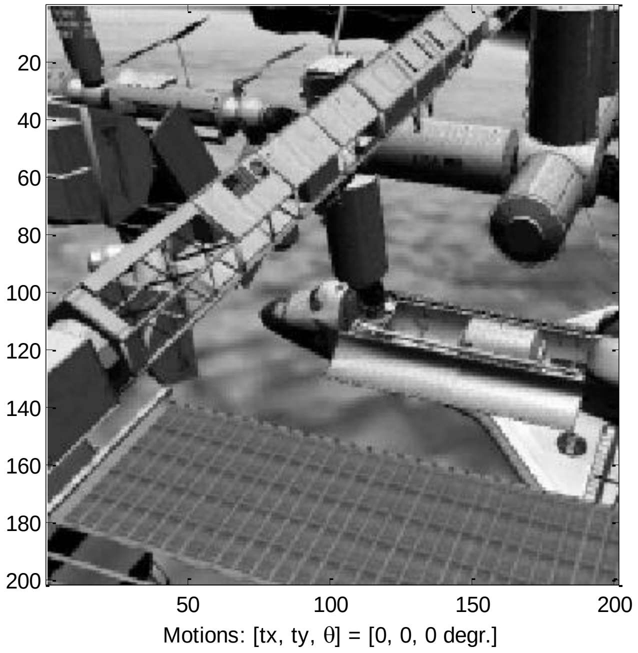


Figure 19. Low Resolution ISS - Shuttle Docking Reference Image

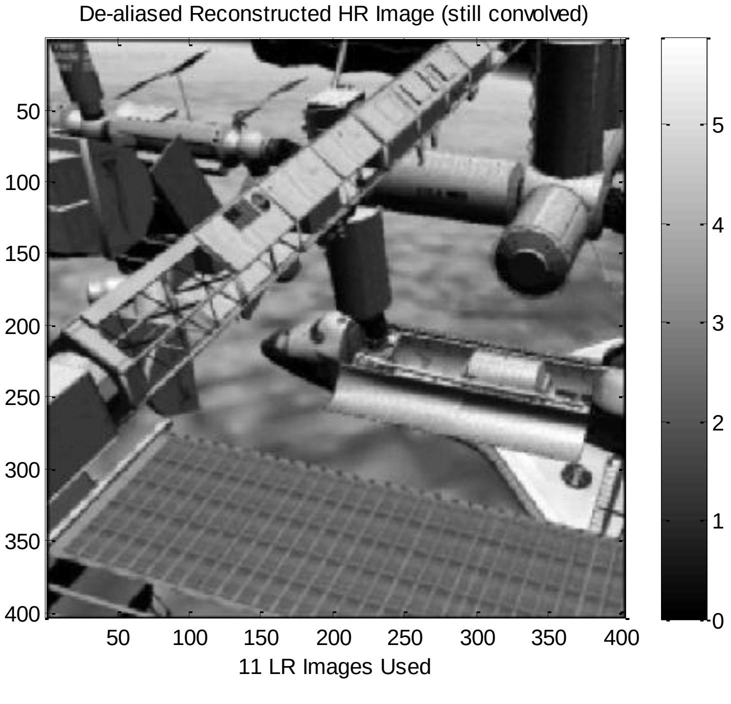


Figure 20. Dealiased Image from 11 Randomly Shifted ISS – Shuttle Docking Images

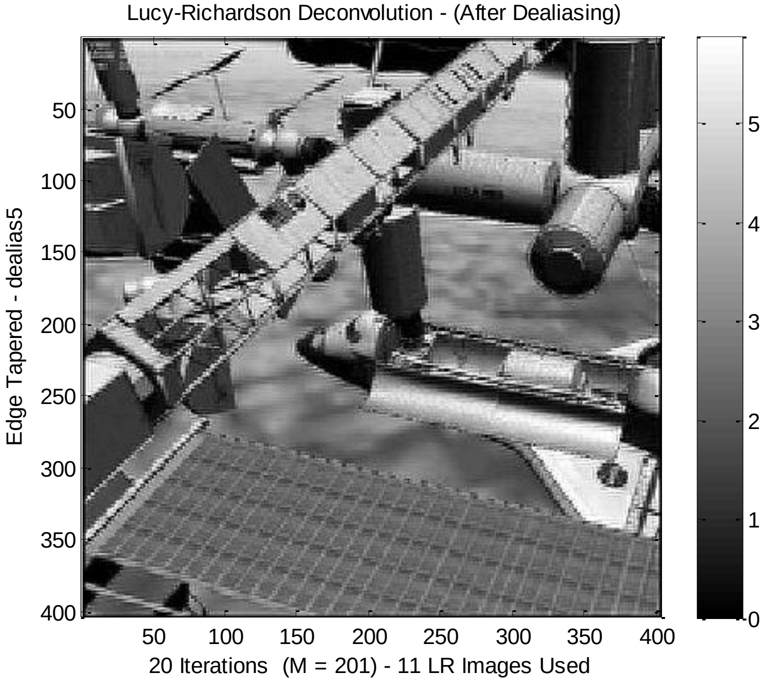


Figure 21. L-R Deconvolution After Dealiasing 11 Randomly Shifted Docking Images

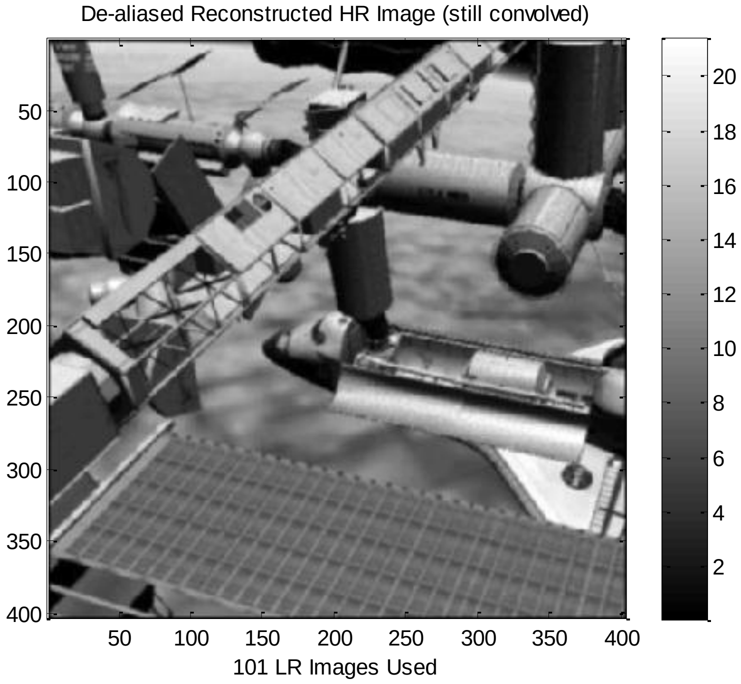


Figure 22. Dealiased Image From 101 Randomly Shifted Docking Images

Approved for public release; distribution unlimited.

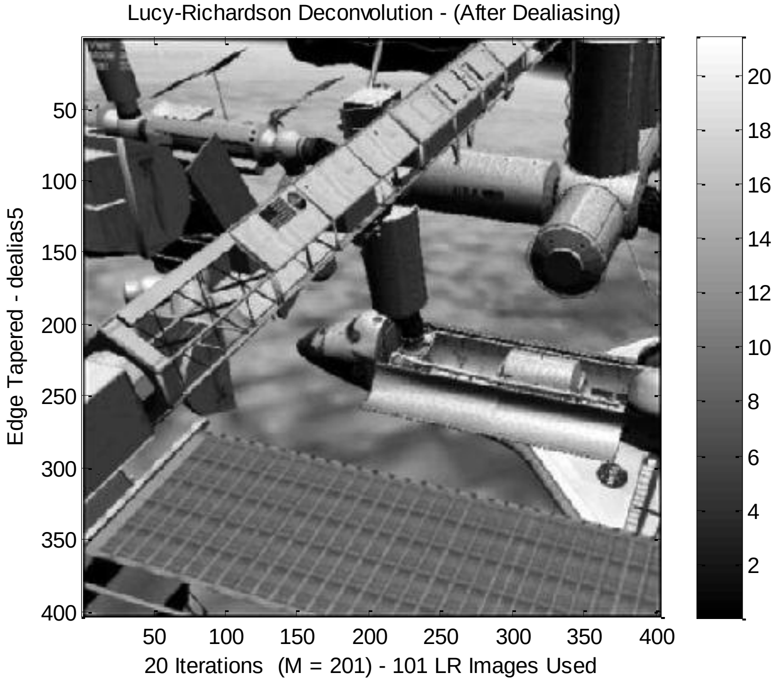


Figure 23. L-R Deconvolution After Dealiasing 101 Randomly Shifted Docking Images

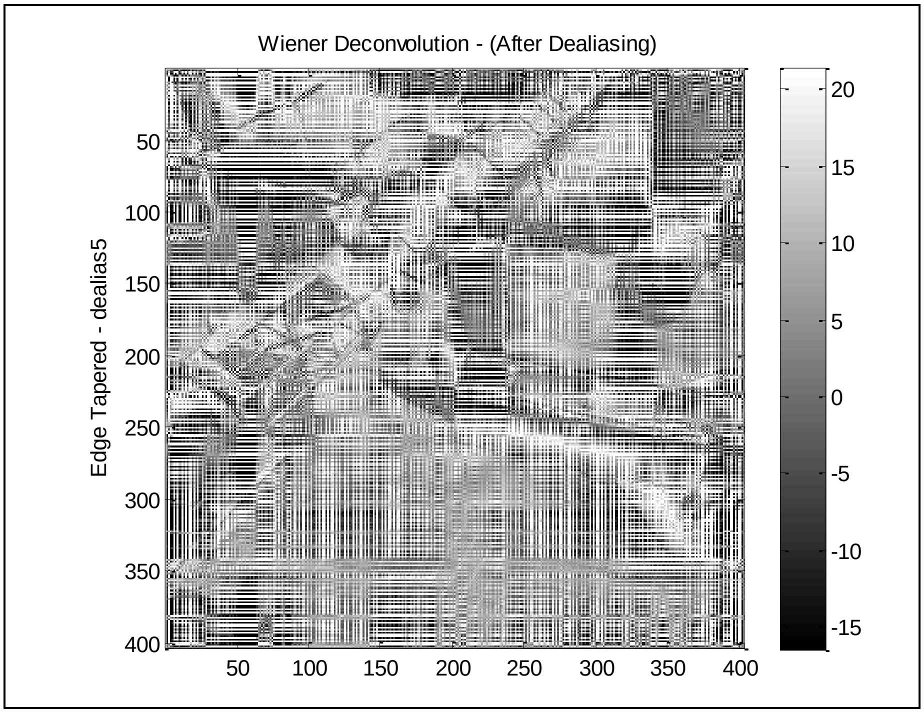


Figure 24. Wiener Deconvolution After Dealiasing 101 Randomly Shifted Docking Images

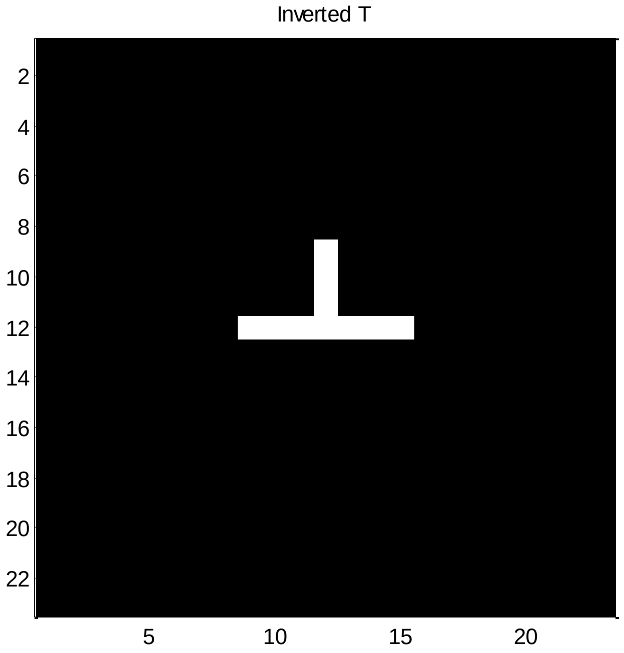


Figure 25. Initial Inverted T Image

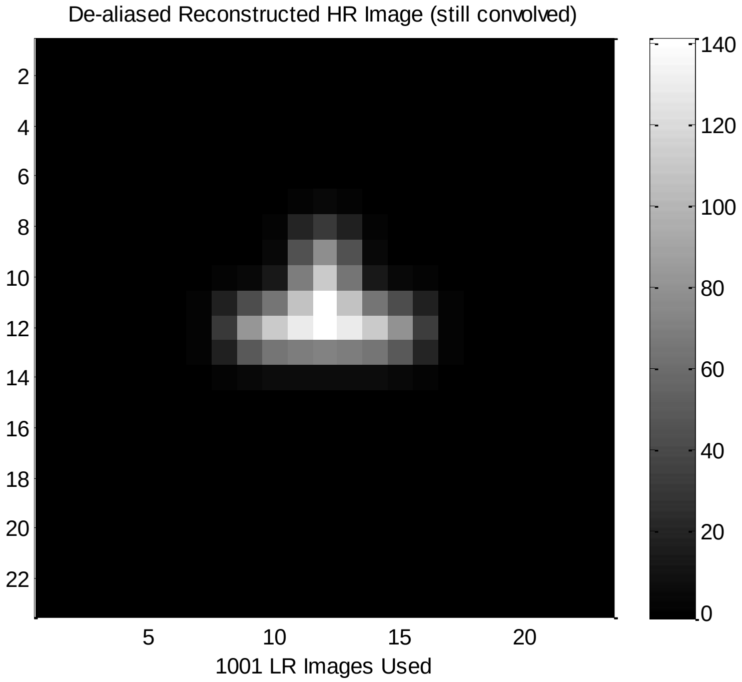


Figure 26. Dealiased Image of 1001 Randomly Shifted Inverted T Images

Approved for public release; distribution unlimited.

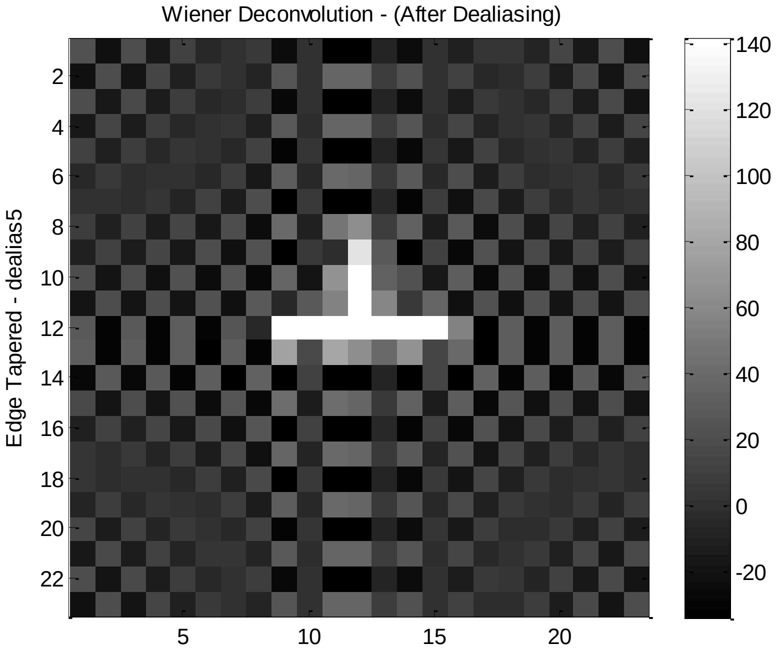


Figure 27. Wiener Deconvolution After Dealiasing 1001 Randomly Shifted Inverted T Images

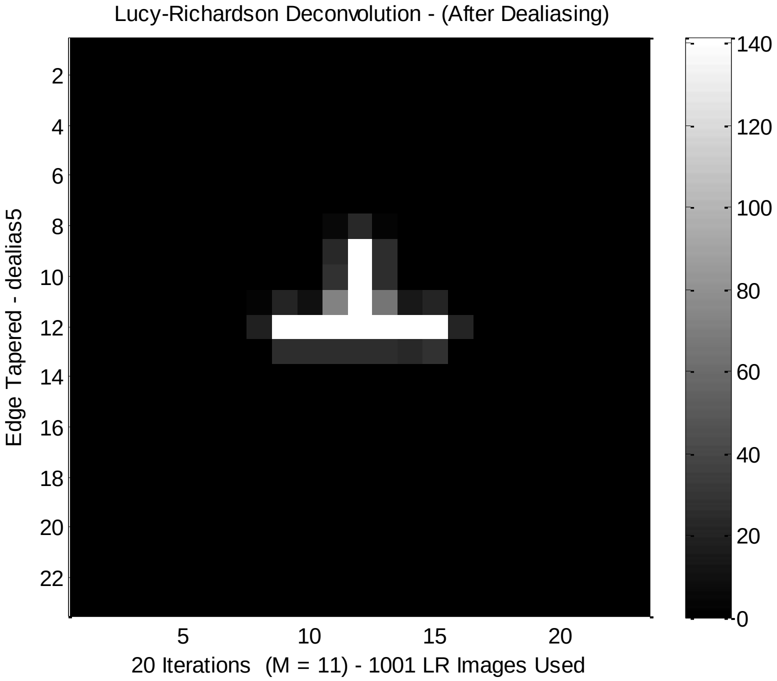


Figure 28. L-R Deconvolution After Dealiasing 1001 Randomly Shifted Inverted T Images

were developed and coded. Perfect reconstruction cannot occur except as the number of displaced images approaches infinity. For the inverted T object shown in figure 25, the dealiasing of 1001 randomly shifted low resolution snapshots of it is shown in figure 26, followed by its Wiener deconvolution in figure 27, and the Lucy-Richardson deconvolution in figure 28. The results show that the Lucy-Richardson deonvolution algorithm was clearly superior to that of Wiener deconvolution, even though the Wiener performed very well for the synthetic binary star example.

An effort was made to develop a more general and practical code for dealiasing images that had been initially both displaced and rotated. The general formulas and logic for calculating the exact overlap area of an irregular polygon formed from overlapped square grids were straightforward, but execution for a multi-frame sequence involving rotations and shifts proved excessively memory- and time-intensive, hence impractical. We subsequently pursued “drizzling” or variable-pixel linear reconstruction (VPLR) as developed in references [5] and [6]. The resulting code was still not efficient enough to be useful. An extensive Google search on the subject revealed numerous correspondences and communications on this subject, but none of them in the context of the Matlab language. At the present time we have been unable to locate an established Matlab script for this purpose and have been forced to put this particular aspect of our research effort aside.

2.3.4 Nonlinear High Resolution Reconstruction

Having obtained the triplet (h,v,θ) for each registered LR image, we pursued the nonlinear HR reconstruction described in reference [2]. This involved minimizing a regularized cost function defined by equation (34) in [2]. A regularized solution to the nonlinear unconstrained cost function minimization was generated using a conjugate-gradient iterative procedure, the convergence of which is indicated by figure 29 for the binary star example. Comparing the HR reconstructed images of figure 30 and its deconvolution shown in figure 31 makes evident the importance and benefits of applying deconvolution.

The weights, $w_{m,k}$, appearing in equation (34) of reference [2] require specification for the HR reconstruction treatment of images having general motions. An efficient formulation or algorithm for so doing is not provided in [2]. This led us to the same problem of specifying overlap area weights encountered in section 5.

2.3.5 Summary of Results for Image Reconstruction

We developed a Matlab script based on the analysis and discussion appearing in reference [2] for multi-frame image registration. It did not lead us to a sufficiently accurate algorithm for predicting sub-pixel shifts and rotations. We traced the problem to the 3-term Taylor series for motion predictions, valid only near the expansion points taken at the centers of the images. The analysis, however, led to matrix elements constructed with pixel intensities spanning the entire image. Those contributions away from the origins severely distorted the accuracy of the predict-

Approved for public release; distribution unlimited.

tions. There was no mention of this issue in [2]. The most effective way we found to compensate for the problem was to develop a weighted linear least-squares formulation for the cost function

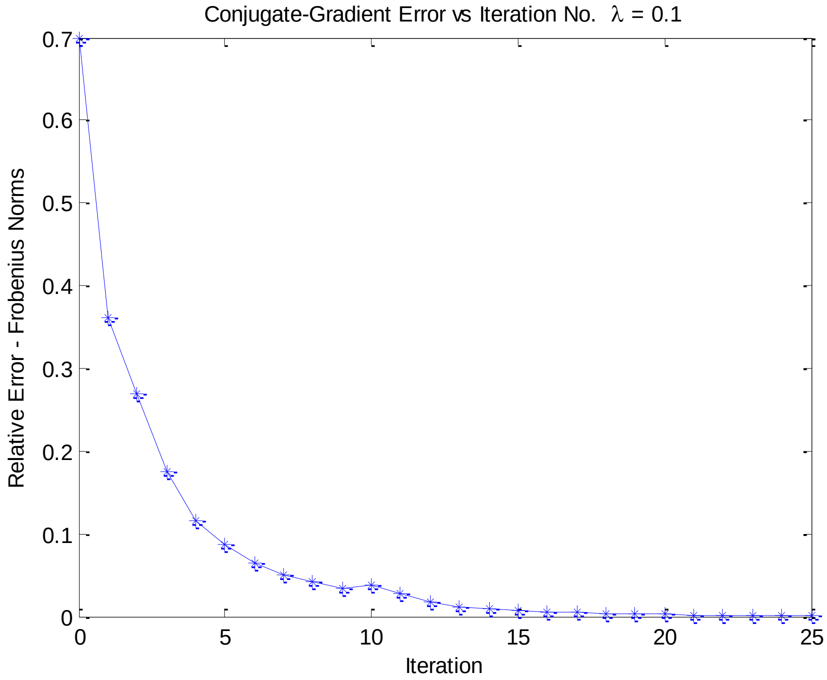


Figure 29. Binary Star Conjugate-Gradient Convergence Plot

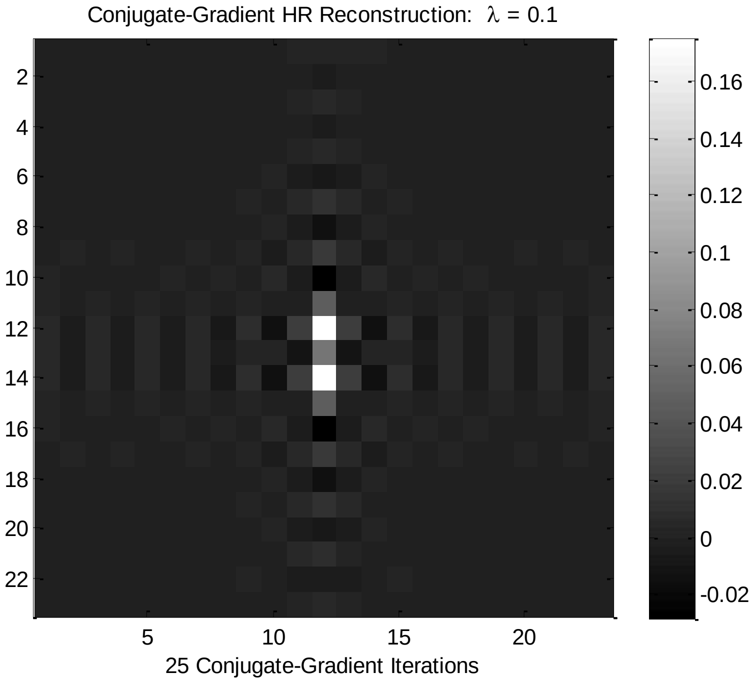


Figure 30. Binary Star Nonlinear HR Reconstruction

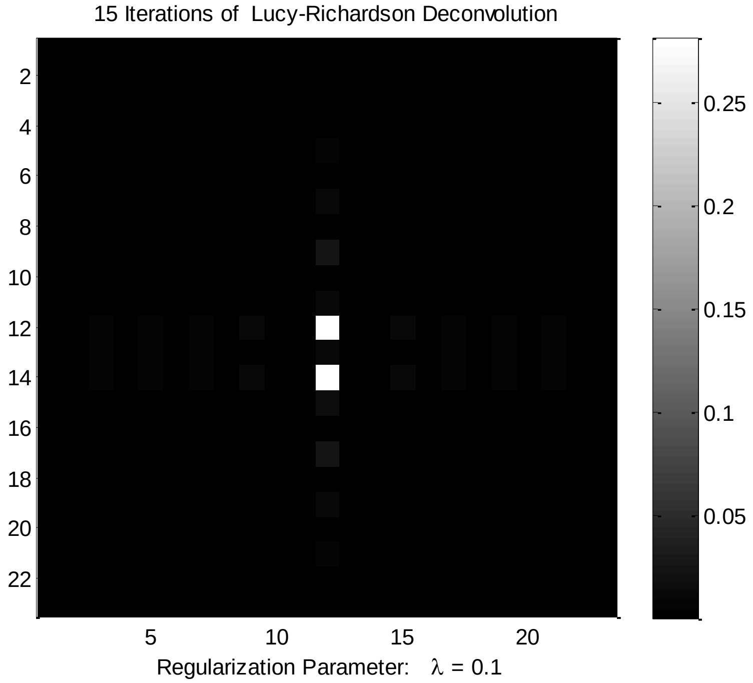


Figure 31. L-R Deconvolution of Nonlinear HR Reconstruction from Binary Star Frames

that emphasized the central region pixel contributions to the matrix elements. Gaussian weight functions centered at the origins led to improvement in the algorithm; however, we found no *a priori* way to determine the optimal width σ of the Gaussian for each image. This led us to use a search algorithm for the optimum σ, the one that gave the minimum $L_1 - norm$ of the average pixel position error. While it gave better accuracy in most cases, the execution time of the algorithm increased to the point where a different algorithm was sought. We subsequently found and modified a function from the SR program of [3] that accomplished our registration objectives. As shown in Tables 1-3, registration for rotations up to almost $5°$ improved considerably as tolerances were tightened, giving rotation predictions to within thousandths of a degree. It appears the script can be improved further both with regard to speed and accuracy.

We pursued both linear and nonlinear HR image reconstructions for general image motions. A key step in both approaches required the definition of weight factors proportional to irregular areas formed by the intersection of LR pixel elements overlapping HR grid pixel elements. The difficulty in determining them efficiently in the reconstruction process was reported above. A "drizzling" algorithm was developed in Matlab as an alternative, but even this required excessive time and memory resources to allow testing with multiple images. With both linear and nonlinear HR image reconstruction methods, we were able to get good results for 2 situations: 1) the exact and ideal case involving 4 half-pixel shifted images; and 2) using, for example, 100 images randomly displaced by sub-pixel amounts. An examination of the literature and internet communications on this subject remains in progress. Our investigations confirmed the importance of deconvolution after dealiasing. In this regard, Matlab's blind deconvolution function failed us whereas the Lucy-Richardson deconvolution algorithm made significant improvement over reconstructions that had previously not been deconvolved.

3. SABER SOFTWARE

This section describes some of the software we have written in order to facilitate the analysis of SABER data. The overall process we engage in involves (1) downloading or otherwise obtaining files containing SABER data that have been processed into physical units by the operations team; (2) producing event data files and storing them in a certain file structure; and (3) accessing those data in ways that allow us to address science questions. Results obtained from the last of these steps have been reported in our Interim Scientific Reports [7, 8, 9, 10, 11] and in a research paper [12].

The standard file type used for SABER data is Network Common Data Form (NetCDF). These are so-called self-describing files, each with a header that specifies the arrangement and contents of the arrays of data it contains. Downloading a large number of large NetCDF files, function (1) above, can be accomplished through FTP, with UNIX scripts to automate the process if necessary. However, researchers working with major portions of the SABER database are encouraged to simply send portable hard drives to GATS, Inc., the NASA contractor responsible for this level of processing. This is the approach we have taken.

The codes we wrote to accomplish function (2) are described below, in this section. The main one is called WRT_EVFS. Codes we wrote to accomplish function (3) have been described previously [10]. They are called READ_SDAT and READ_SL2A. The file structure that they access was also given at that time [10], along with a detailed discussion of files required for them to run. Relevant portions of that discussion are repeated below, and reference is made to tables and figures found in [10].

The software referenced in this report, and in reference [10], has been delivered in the form of source code to the Air Force Research Laboratory.

As noted, the remainder of this section is devoted to function (2), the process we use to write SABER data event files into the appropriate subdirectories. The standard procedure for doing that, described below, is to (a) create a directives file; (b) run a code called MAK_NC_SCR; (c) create whatever new directory structure is needed; (d) run WRT_EVFS; (e) insert a new segment into the index file [10]; and (f) make a backup of the event files just produced. First we give a brief introduction to the instrument. Then we review the file structure we use. We describe the directives file needed to run the codes MAK_NC_SCR and WRT_EVFS, and then the codes themselves, their objectives, and usage. WRT_EVFS is the principal code. Its purpose is to access SABER NetCDF files directly and excerpt their contents to write the event data files. MAK_NC_SCR has a supporting role. It produces three ASCII files, two of which are converted into executables and used as UNIX scripts. Finally, we describe the event files that are produced.

3.1 SABER Instrument

SABER [13] is a 10-channel broadband radiometer, one of four instruments aboard the TIMED satellite. TIMED is an acronym for Thermosphere Ionosphere Mesosphere Energetics and Dynamics; SABER stands for Sounding of the Atmosphere by Broadband Emission Radiometry. Since early in 2002, this instrument has been continuously scanning the limb between the ground and tangent heights of approximately 300 km, recording IR and NIR emissions from CO_2, H_2O, O_3, NO, OH, and O_2. The radiance recorded in each of the channels is reported to the scientific community as so-called Level 1B data. The SABER automated data processing facility also reports, as Level 2 data, many quantities derived from the radiance. Level 2A data are also available; they are a subset of Level 2. These are all available as NetCDF files. The most important quantities specify the atmospheric state, e.g. the temperature, pressure, and constituent densities, over as great a range of altitudes as possible. Other derived products include volume emission rates, energy deposition rates, and cooling rates. All of them support the TIMED objective, which is to understand the energetics and dynamics of the middle atmosphere.

3.2 File Structure and Naming Conventions

SABER's longevity and high duty cycle ensure a prodigious flow of data. These are organized by year, day, orbit and event. An event comprises one full sweep of the scanning mirror, down or up, through the full accessible range of tangent heights. There are about 100 events per orbit, and each NetCDF Level 2 or Level 2A file (L2 or L2A) contains data from an entire orbit. The orbital period is approximately 90 minutes so there are 14 or 15 orbits per day, the beginning of each set by convention at the ascending equator-crossing. Universal time is used, so days begin at zero hours GMT. Complete orbits are assigned to the day during which they begin.

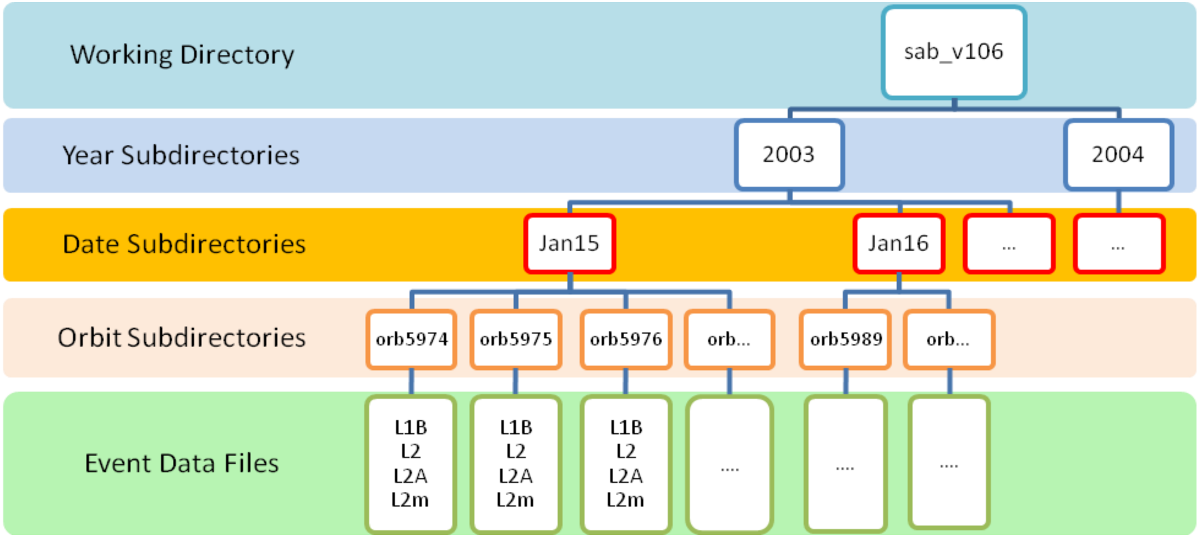


Figure 32. Heirarchy of Subdirectories Required for SABER Data Processing

In our data handling system, running in a UNIX or LINUX environment, the file structure is also organized by year, day, and orbit. Figure 32 illustrates the hierarchy, which is also discussed in reference [10]. At the top of the structure is a working directory, where executables and certain input files (or links to them) reside. Below that are “year”, “day”, and “orbit” subdirectories. The working directory contains executables, the index file, and directives files, or links to them, among other things. It also contains the year subdirectories, and it could contain NetCDF files. Year subdirectories contain date subdirectories. Date subdirectories contain orbit summary files and orbit subdirectories. Orbit subdirectories contain event data files. The purpose of the codes discussed here is to locate and read the binary NetCDF files, and write certain of their contents into ASCII event data files in the orbit subdirectories, the lowest levels shown in figure 32.

The naming conventions are as follows. Level 2 NetCDF filenames have the general form `SABER_L2_yyyyddd_MMMMM_vv.vvXXX.nc`, where “`yyyy`” is a four-digit year, “`ddd`” is a three-digit day of the year (e.g., 001-365), “`MMMMM`” is a five-digit orbit number with leading zeroes if necessary, and “`vv.vvXXX`” is a data product version number. The latter is usually just a 5-digit string like “01.07”—in this case indicating release 1, version 7—in which case “`XXX`” would be a null string. However, the general case allows for an arbitrary appendage, which could be a single character or an extended string. A file `SABER_L2_2002040_00944_01.07.nc` would contain all L2 data of orbit 944, on 09 February 2002, which was very early in the TIMED mission. Level 2A NetCDF filenames are exactly the same, but for an additional “`A`”.

NetCDF files may be located in the working directory, but usually it is convenient to put them elsewhere. Paths to the working directory must then be specified as part of the filenames when referring to them.

The year subdirectories reside directly beneath the working directory, and the date directories below the year. These have four- and five-digit names in the form `yyyy` and `Mondd,` respectively. Upper- and lower-case characters are required for the month in the exact manner shown in figure 32. Orbit subdirectories below the date subdirectories have the form `orbLLLL`. “`LLLL`” is the 3-, 4-, or 5-digit orbit number, in this case with <u>no</u> leading zero.

The L2 data event files themselves have the form `tk_l2_orbMMMMM_ scan_N.dat`. For L2A files, “`l2a`” substitutes for “`l2`”. In either case, “`MMMMM`” is a <u>five</u>-digit orbit number here, with leading zeroes if necessary, but “`N`” is a one-, two-, or three-digit event number.

Putting this together, the full path and filename for a L2 event data file, starting in the working directory, might be `2002/Feb10/orb944/tk_l2_orb00944_scan_1.dat`, referring to event (scan) #1. There normally would be approximately 100 such files in `orb944`, and perhaps an equivalent number of L1B (radiance) event files. `Feb10` would contain 13 or 14 other “orb” directories with similar contents, and `2002` would contain other dates as well.

Note, in reference [10], where a figure similar to figure 32 appears, we refer to “group” subdirectories rather than to “year” and “day” subdirectories. Group subdirectory names are limited to ten characters in the system described there. Including the "/" separator as one character, we can write them in the form yyyy/Mondd, which is what we choose to do. So "group" actually refers to two levels there, a distinction of no consequence in UNIX or LINUX.

We also note that the working directory is associated with a particular data version. When a new version is released, which happens occasionally, an entirely new data structure is built. Orbit subdirectories at the bottom are populated only with data of the current version. This is necessary to avoid ambiguous identification of event data files, and to prevent unintended data comparisons across different versions.

3.3 Directives File

The two codes described below use a single input file, a "directives file" having the purpose of telling the codes what to do. The file is called netc_names. For the most part, it is simply a list of the NetCDF input files to be processed. A sample version is shown in figure 33.

The first record is a character string with certain requirements. The first four characters give the year. Somewhere before the 50th character, either "L2" or "L2A" should appear, to distinguish the two file types. One cannot process both types at the same time. Nor can one process files from different years in the same run.

For cases in which NetCDF filenames include extended strings in the data versions—e.g., when "XXX" in the filename is not null—an option is available to rename the data version with a simpler string when output is written. If "vv.vvXXX" consisted of a long string with, for example, processing parameters included, one could use a shorter string to replace it. To do this, the first record of the directives file must include (somewhere) the substring "v1.". In such a case, whatever five characters follow the "v" are used to construct an abbreviated version identifier, possibly something like "1.07a". This new identifier is written into the event data filenames. If that string is not found, the original data product version is used.

The second record is an output filename root. It is used to construct output names for various files that the codes produce—not including the event data files.

Beginning at the third record, the directives file consists of a list of NetCDF files, including path and filename, for all orbits that are intended to be processed. No more than 1000 such files can be done at once. Such a list is easily constructed using simple UNIX commands ("ls"). It is not necessary that all files on the list reside in the same storage directory. It is necessary that the list be ordered according to increasing orbit number.

The first few records of a sample directives file are given in figure 33. The NetCDF filenames have a compact version number, 1.07, so no override is found on the first record. The output filename root, MM2006_v1.07_, includes our mnemonic for files from the SABER yaw cycle from March to May 2006. The path, /st_netc/SABER_Level2/2006/077, gives the lo-

Approved for public release; distribution unlimited.

cation, including subdirectories, where these NetCDF files are found, a directory with files from day 77 (March 18) of 2006. Filenames themselves begin with “SABER_L2...” in all cases.

```
2006   L2
MM2006_v1.07_
/st_netc/SABER_Level2/2006/077/SABER_L2_2006077_23144_01.07.nc
/st_netc/SABER_Level2/2006/077/SABER_L2_2006077_23145_01.07.nc
/st_netc/SABER_Level2/2006/077/SABER_L2_2006077_23146_01.07.nc
/st_netc/SABER_Level2/2006/077/SABER_L2_2006077_23147_01.07.nc
.....
```

Figure 33. First Records of a Sample Directives File, `netc_names`

3.4 Program MAK_NC_SCR

This code produces several ASCII output files. It locates netc_names in the working directory and reads the list of NetCDF files. Each of those names encodes the pertinent year, day of year, and orbit, from which MAK_NC_SCR constructs a list of dates and orbits for which new subdirectories need to be created. Its first output file consists of ASCII strings consisting of mkdir commands, e.g. UNIX statements for producing date subdirectories and, beneath them, orbit subdirectories. (Our convention is that the year subdirectory above them should already exist.) The output filename will use the filename root with the appendage “nc_mak_dir.scr” and will contain separate records for each new date, and also for each orbit encountered. It is necessary only to make that file executable and run it as a UNIX script to produce all the empty directories and subdirectories needed subsequently. Figure 34 gives excerpts from such a file.

```
mkdir 2006/Mar18
mkdir 2006/Mar19
mkdir 2006/Mar20
mkdir 2006/Mar21
....
mkdir 2006/May20
mkdir 2006/Mar18/orb23144
mkdir 2006/Mar18/orb23145
mkdir 2006/Mar18/orb23146
....
```

Figure 34. Excerpts from a Script for Creating Directory Structure

MAK_NC_SCR also generates a set of commands that can be used to back up the event data files once they are created. For each orbit, there is a record consisting of the command for creating a g-zipped tarfile containing the complete set of L2 (or L2A) files in the orbit directory. As with the script for making date and orbit directories, this is intended to be run as a script from the working directory. Its name will be the root with the appendage “nc_wrt_tar.scr”. Figure 35 gives the first few records of such a file.

Finally, MAK_NC_SCR writes an output file to be inserted into an existing index file. The index file, named `sdat_index`—described in referencee [10] and illustrated there in figure 2— specifies the location of event files for every orbit. It is needed for the data processing codes READ_SDAT and READ_SL2A. `sdat_index` is an ordered list of orbits or orbit ranges [10] and MAK_NC_SCR produces whatever segment of that file may be needed for the newly-processed orbits. The name of this file is the root with the appendage “`nc_index_file`”.

```
cd 2006/Mar18
tar cfz tk_l2_orb23144_v1.07_tar.gz orb23144/tk*
tar cfz tk_l2_orb23145_v1.07_tar.gz orb23145/tk*
tar cfz tk_l2_orb23146_v1.07_tar.gz orb23146/tk*
tar cfz tk_l2_orb23147_v1.07_tar.gz orb23147/tk*
tar cfz tk_l2_orb23148_v1.07_tar.gz orb23148/tk*
tar cfz tk_l2_orb23149_v1.07_tar.gz orb23149/tk*
tar cfz tk_l2_orb23150_v1.07_tar.gz orb23150/tk*
tar cfz tk_l2_orb23151_v1.07_tar.gz orb23151/tk*
tar cfz tk_l2_orb23152_v1.07_tar.gz orb23152/tk*
tar cfz tk_l2_orb23153_v1.07_tar.gz orb23153/tk*
tar cfz tk_l2_orb23154_v1.07_tar.gz orb23154/tk*
tar cfz tk_l2_orb23155_v1.07_tar.gz orb23155/tk*
tar cfz tk_l2_orb23156_v1.07_tar.gz orb23156/tk*
tar cfz tk_l2_orb23157_v1.07_tar.gz orb23157/tk*
cd ../..
cd 2006/Mar19
tar cfz tk_l2_orb23158_v1.07_tar.gz orb23158/tk*
tar cfz tk_l2_orb23159_v1.07_tar.gz orb23159/tk*
....
```

Figure 35. Segment of a Script for Creating Backup Files

3.5 Program WRT_EVFS

NetCDF codes and interfaces have been developed and distributed by Unidata, a program operating through UCAR (University Corporation for Atmospheric Research) having the mission to “provide the data services, tools, and cyberinfrastructure leadership” to benefit the Earth Sciences community. Software packages, including source code and libraries required to work with NetCDF files, are available at www.unidata.ucar.edu/software/netcdf. Extensive documentation, including instructions for installing the packages, descriptions of callable routines, and tutorials, is also available on that website.

There are specific NetCDF routines for all permissible operations, and all of them are accessed as FORTRAN functions. For example, we open NetCDF files by

```
IRTV = NF_OPEN(FLNM,NF_NOWRITE,NCID)
```

where FLNM is the name of the file and NF_NOWRITE (or any variable with a value set to zero or .FALSE.) specifies read-only. The last argument, NCID, is a specific ID number used whenever the file is subsequently referenced, and IRTV acquires the value zero when the operation is successful. Other functions are used in an analogous fashion. In particular, for each variable to be read, a variable ID for that particular variable has to first be extracted, to be used whenever the variable is subsequently referenced. For the SABER event number, this is accomplished by

```
IRTV = NF_INQ_VARID(NCID,'event',IDEV)
```

where NCID is the file identifier obtained from NF_OPEN and IDEV is the ID for that variable. Then, whenever one of the NF_GET functions is invoked to extract numerical data, variable IDs like IDEV are required input for the module that is called.

The NetCDF package provides a utility, ncdump, for quickly reading any NetCDF file. This is generally not a useful way to extract large volumes of data. However, using the -h parameter limits its output to a simple list of the elements of the file, rather than the full numerical content. In the present instance, those elements are dimensions, variables, and global attributes. One can thereby quite easily find what all the variable names are, e.g. event in the example above.

Making a successful executable from FORTRAN source has two simple requirements. The first is to link the NetCDF library, libnetcdf.a. The other is to have the INCLUDE file provided, called netcdf.inc, available in the directory where the executable is being built. That is needed to specify data types, fill values, error codes, default values, and externals.

Our main code for writing event files is WRT_EVFS. There are separate versions of this code for L2 and for L2A. In each case, comments in the source code include a list of the names and dimensions of all variables extracted, as well as the variable ID, the FORTRAN name assigned to the variable, and its type. The code extracts a fairly small subset of everything in the NetCDF files, but the set could easily be expanded with a few extra calls. The main problem would then be how to write out all results in manageable form.

As noted earlier, WRT_EVFS uses the same directives file as MAK_NC_ SCR. It actually ignores quantities on the first record of that file unless the version identifier needs to be overridden. While running, it writes a progress list of successfully-opened NetCDF files to standard output. It also writes a log file, having a name consisting of the output filename root (the second record) with the appendage “netc_run.log”. This reports the number of events processed for each orbit, and the number of events missing between the first and last event in the orbit.

3.6 Event Date Files

Our L2 and L2A event files report relatively small subsets of all the information found in the NetCDF files. They consist of a header with 11 records, followed by columns of data arranged by altitude (highest altitude first). Altitudes are the same as the interleaved SABER tangent-point grid, having a spacing of approximately 0.4 km. The first three columns give the altitude and the retrieved temperature and pressure, and are identical for L2 and L2A. In fact, the main difference between those file types is that our L2 event files report products not found in the L2A NetCDF files. Table 5 lists the contents of these files.

Figure 36 shows a L2 file header, for SABER orbit 23144, scan #10. It includes, among other things, the experiment date, orbit number, event number, processing dates and some solar/geomagnetic data for the time in question. It also specifies the time (as UT), solar zenith angle (SZA), and latitude and longitude of the tangent point. These latter quantities vary slowly during the time it takes to complete an event (a mirror sweep). Headers therefore give their values at those moments when tangent points are nearest to 40, 80, 120, and 200 km.

```
C SABER L2 day = 2006077 = 18-Mar-06                    Orbit # = 23144  scan = 10
C NET-CDF File Generated = 2007330150114                        L2 version =  1.07
C Data written with code WRT_EVFS, version 5 Event-file Date = 04-27-2011  15:58:49
C Solar Ap = 30  Solar Kp =   1.7     Solar F10.7 =  72.40    Solar F10.7a = 150.00
C Universal times are   1.7459   1.7446   1.7433   1.7404
C Tangent points (km) are  40.16   80.12  120.16  199.84
C Solar Zenith Angles are 137.00  137.62  138.25  139.58
C Latitudes  (degs N) are  25.83   26.06   26.29   26.73
C Longitudes (degs E) are  12.39   11.47   10.51    8.47
C Event orbital information: Down   Asc   Boresight-Moon separation angle =  68.11
C tp(km) Temp(K)   Press(mb)   O vmr     CO2 vmr   O_1D vmr   O3-96 vmr    H vmr
```

Figure 36. Level 2 Event Data File Header

The header also includes the worst-case separation angle between the SABER boresight and the Moon during each event, plus the tangent height at which it occurs. The purpose of including that information is to enable screening for events whose radiance profiles are contaminated by moonlight. Fortunately, in recent releases, such events are now reliably removed prior to the production of NetCDF files, making that information superfluous for normal processing.

Note, in figure 36 the last two lines, those with the separation angle and the data column headers, have been truncated on the right.

The range for most data reported in L2 or L2A event files is restricted to altitudes between about 0 and 150 km. Indeed, the temperature/pressure retrievals usually do not extend even that high, nor do reports for other products that depend on them. Fill data (-999) are used wherever results cannot be validated.

One product that is quite useful above 150 km is the nitric oxide (NO) volume emission rate (VER). During geomagnetic storms or other excited periods, it is a viable quantity well above the E region of the ionosphere. Our SABER event files record this using a supplemental (“top”) altitude grid that covers the range from 150 km to ~270 km, with a corresponding set of VERs. This is in addition to the regular range ~0-150 km. The last two columns of the L2 and L2A event files are reserved for these quantities.

Table 5. Summary of Contents and Format, SABER Level 2 and 2A Event Data Files

Quantity	L2 Column	L2 Format	L2A Column	L2A Format	Comments
Altitude (km)	1	F6.2	1	F6.2	z < 150 km
Temperature (K)	2	F10.3	2	F10.3	
Pressure (mb)	3	E11.3	3	E11.3	
Atomic oxygen vmr	4	E11.3	4	E11.3	From OH and O_3
CO_2 vmr	5	E11.3	5	E11.3	Retrieved in daytime only
$O(^1D)$ vmr	6	E11.3	6	E11.3	Retrieved in daytime only
Ozone vmr	7	E11.3	7	E11.3	From 9.6 µm emission
Atomic hydrogen vmr	8	E11.3	8	E11.3	
Nitric oxide VER	9	E11.3	9	E11.3	z <150 km
Hydroxyl VER (OH-A)	10	E11.3	10	E11.3	From 2.0 µm emission
Hydroxyl VER (OH-B)	11	E11.3	11	E11.3	From 1.6 µm emission
Atomic oxygen vmr	12	E12.3	-	-	Input; O_ext
Ozone vmr	13	E11.3	-	-	From 1.27 µm emission
Altitude	14	F12.2	12	F12.3	z > 150 km (“top”)
Nitric oxide VER	15	E11.3	13	E11.3	z > 150 km (“top”)

To summarize, WRT_EVFS places all the event data files into the orbit subdirectories, where the other data-processing codes [10] expect to find them. Once the index file is updated with the new segment produced by MAK_NC_SCR, the data processing code READ_SL2A can produce orbit summary files [10] from their headers and then proceed to whatever analysis is called for.

4. SYNOPSIS OF INTERIM REPORTS

Much of the scientific effort conducted under the provisions of our contract focused upon conditions in the upper stratosphere, the mesosphere and the lower thermosphere, the region TIMED was designed to explore. Because of the high-inclination orbit, and the near-continuous duty cycle of SABER, multiple repeated observations became newly available, making it possible to study the region with much more detailed information than ever before. This section very briefly summarizes some of our work, which has been reported in more detail in the past.

Using an early version of SABER retrieved temperature, we studied global characteristics of the mesopause (temperature, altitude) at solstice [7]. This confirmed and refined the then-evolving notion of a global two-level mesopause, e.g. a mesopause with a mean height of ~86 km in the summer hemisphere and a mean height close to 100 km in the equatorial regions and the winter hemisphere. It showed that the transition between the two altitude regimes, which occurs near ~20° latitude in the summer hemisphere, is very abrupt. On the other hand, the mean temperature —in the ~170-180 Kelvin range on the “winter” side—changes gradually, over a latitude range

Approved for public release; distribution unlimited.

of ~30°, toward its minimum of ~130 K near the summer pole. Above and beyond establishing the mean state, these data give an unprecedented view of the variability of the region. Polewards of ~60 ° in the summer, the distribution of mesopause altitudes about the mean is quite narrow, only ~1-2 km. But in the "winter" regime it broadens considerably, possibly reflecting a stronger influence of planetary, tidal, and gravity waves. In fact, in this regime temperature inversion layers (TILs) are quite common in the mesosphere, particularly in regions and at local times with strong tidal influence. We showed that TILs affect the mesopause characteristics, and characterized their occurrence distributions in latitude and local time [7].

We also studied the properties of mesospheric bores [7], by numerically solving the differential equations governing their propagation modes. Using impulsive forcing with measured wind and temperature profiles, we determined the vertical displacement for altitudes within the duct as a function of time and horizontal distance, and thus the evolution of the wave. The calculations reproduce the phase speed of an observed bore to a good approximation.

Subsequently [8], we looked at the SABER CO_2 infrared limb radiance database in the region just above the mesopause, where a positive radiance gradient is commonly but not consistently a feature of vertical profiles of longwave (15 µm) emissions. We studied this feature as a function of latitude, local time, and season, and with the help of a global tidal model and our own non-LTE codes were able to show that tidal perturbation of the temperature field in the lower thermosphere is directly responsible for observed variability in the radiance.

Our examination of other portions of the SABER database [9] revealed extraordinary conditions in the polar winter mesosphere in early 2004 and 2006, years during which very strong stratospheric sudden warmings (SSWs) occurred. The OH layer was unusually low—as much as 8 km below the nominal altitude of 87 km—and very bright in this region. The temperature structure of the entire middle atmosphere was also perturbed; a temperature minimum was found in the 40-50 km region and an "elevated stratopause" formed near 80 km. We quantified these anomalous effects, studied their evolution over periods approaching two months, and contrasted them with what occurred in the "normal" years of 2003 and 2005. We concluded that greatly enhanced downward transport from the mesopause region or above was responsible for anomalous OH emissions, and elaborated upon it in a research paper [12]. Subsequently, in 2009, these unusual conditions occurred once again following a strong SSW.

Other than 2005, every northern winter between the launch of TIMED and 2009 featured one or more SSWs. In 2008 alone there were four, not all major warmings, but the three winters mentioned above were the only ones to produce an elevated stratopause and extraordinary OH emissions. So we made a systematic study [11], using eight SABER northern winters, the objective being to see what changes (above and beyond what we had already discerned) the SSWs might have effected in the mesosphere. We found that mesospheric cooling accompanied each major SSW and most minor ones, and we quantified the altitude ranges and durations over which it occurred. When cooling could be seen in the upper mesosphere, it preceded the SSW and the cooling in the lower mesosphere; for those cases cooling appeared to descend at a rate of ~5 km/day. We also observed a rise of ~2-3 km in the OH layer altitude at the time of the SSWs in 2006 and 2009, prior to its rapid descent in the week following. (The 2004 SSW occurred before the TIMED yaw to the north, so the only observations available that year came in its aftermath.) For other years, no such initial rise was detected. However, modest but significant decreases in layer altitude did occur following most SSWs. One of the conclusions of this study is that SSWs produce a continuum of effects in the mesosphere; indeed, that wave-induced perturbations of the stratosphere that do not necessarily qualify as SSWs may nonetheless affect higher-altitude regions in similar ways. Evidence for this comes from SABER data in 2010, not included in reference [11]. That was a year without an "official" SSW, but there was a substantial warming in the upper stratosphere and, in the weeks following, a distortion of the mesospheric temperature structure—not as complete a distortion as in 2004, 2006, and 2009 but reminiscent of it in many respects, anyway.

We also investigated suggestions about stratospheric warming and mesospheric cooling in tropical zones in the aftermath of strong SSWs, but were unable to substantiate such effects [11].

We have also reported in the past on the two main codes we and others use to perform these studies using SABER data [10]. That report also includes a few examples of the results we have obtained.

REFERENCES

[1] Khare, K. and George, N., "Sampling theory approach to prolate spheroidal wavefunctions," *Journal of Physics A: Mathematical and General*, **36**, 2003, pp. 10011-10021.

[2] Hardie, R., Barnard, K., Bognar, J., Armstrong, E., and Watson, E., "High-resolution image reconstruction from a sequence of rotated and translated frames and its application to an infrared imaging system," *Optical Engineering*, **37**(1), Jan 1998, pp. 247-260.

[3] Vandewalle, P. and Krichane, K., "keren.m," Laboratory of Audiovisual Communications (LCAV), Ecole Polytechnique Federale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland, 2005-2007.

[4] Zalevsky , Z. and Mendlovic, D., **Optical Superresolution**, Springer Verlag, New York, 2003.

[5] Fruchter, A., Hook, R., "Drizzle: A Method For The Linear Reconstruction of Undersampled Images," *Publication of the Astronomical Society of the Pacific*, **114**, 792, pp. 144-152, 2001.

[6] Merino, T. and Nunez, J., "Super-Resolution of Remotely Sensed Images With Variable-Pixel Linear Reconstruction," *IEEE Trans. On Geoscience and Remote Sensing*, **45**, 5, May 2007, pp. 1446-1457.

[7] Wintersteiner, P., and Cohen, E., "Observations and Modeling of the Upper Mesosphere: Mesopause Properties, Inversion Layers, and Bores," AFRL-VS-HA-TR-2005-1162, Arcon Corp., Waltham, MA, October 2005.

[8] Wintersteiner, P., "Observation of Tidal Effects on LWIR Radiance Above the Mesopause," AFRL-VS-HA-TR-2007-1034, Arcon Corp., Waltham, MA, March 2007.

[9] Wintersteiner, P., "Anomalous Conditions in the Middle Atmosphere During Boreal Winters of 2004 and 2006," AFRL-RV-HA-TR-2008-1006, Arcon Corp., Waltham, MA, December 2007.

[10] Wintersteiner, P. and Cohen, E., "Utilizing Middle Atmospheric Data from SABER: Studies for Possible Improvements in Optical Imagery using Superresolution," Interim Scientific Report #4, Arcon Corp., Waltham, MA, November 2008.

[11] Wintersteiner, P., "Sounding of the Atmosphere using Broadband Emission Radiometry (SABER) Observations of Polar Winter Conditions in 2009; Comparisons with Years 2002-2008," AFRL-RV-HA-TR-2010-1116, Arcon Corp., Waltham, MA, May 2010.

[12] Winick, J.R., Wintersteiner, P.P., Picard, R.H., Esplin, D., Mlynczak, M.G., Russell, J.M. III, and Gordley, L.L., "OH layer characteristics during unusual boreal winters of 2004 and 2006," *J. Geophys. Res.*, **114**, A02303, doi:10.1029/2008JA013688, 2009.

[13] Russell, J.M. III, Mlynczak, M.G., Gordley, L.L, Tansock, J., and Esplin, R., "An overview of the SABER experiment and preliminary calibration results," *Proc. SPIE*, **3756**, pp. 277-288, 1999.

DISTRIBUTION LIST

DTIC/OCP
8725 John J. Kingman Rd, Suite 0944
Ft Belvoir, VA 22060-6218 1 cy

AFRL/RVIL
Kirtland AFB, NM 87117-5776 2 cys

Official Record Copy
AFRL/RVBYB/Peter Crabtree 1 cy

Approved for public release; distribution unlimited.